UNIVERSITY OF
ILLINOIS LIBRARY
AT URBANA-CHAMPAIGN
ENGINEERING

JUN 20 1980

The person charging this material is responsible for its return to the library from which it was withdrawn on or before the **Latest Date** stamped below.

Theft, mutilation, and underlining of books are reasons for disciplinary action and may result in dismissal from the University.

To renew call Telephone Center, 333-8400

UNIVERSITY OF ILLINOIS LIBRARY AT URBANA-CHAMPAIGN

ENGINEERING

PHOTO REPRODUCTION DEC 11 REC'D

MAR 16 REC'D INTERLIBRARY LOAN

MAY 15 REC'D

PHOTO REPRODUCTION

JUN 3 1988

APR 2 REC'D

MAY 16 REC'D

PHOTO REPRODUCTION

MAY 14 REC'D

PHOTO REPRODUCTION

SEP 6 REC'D

INTERLIBRARY LOAN

PHOTO REPRODUCTION

NOV 17 REC'D

SEP 22 REC'D

Interlibrary

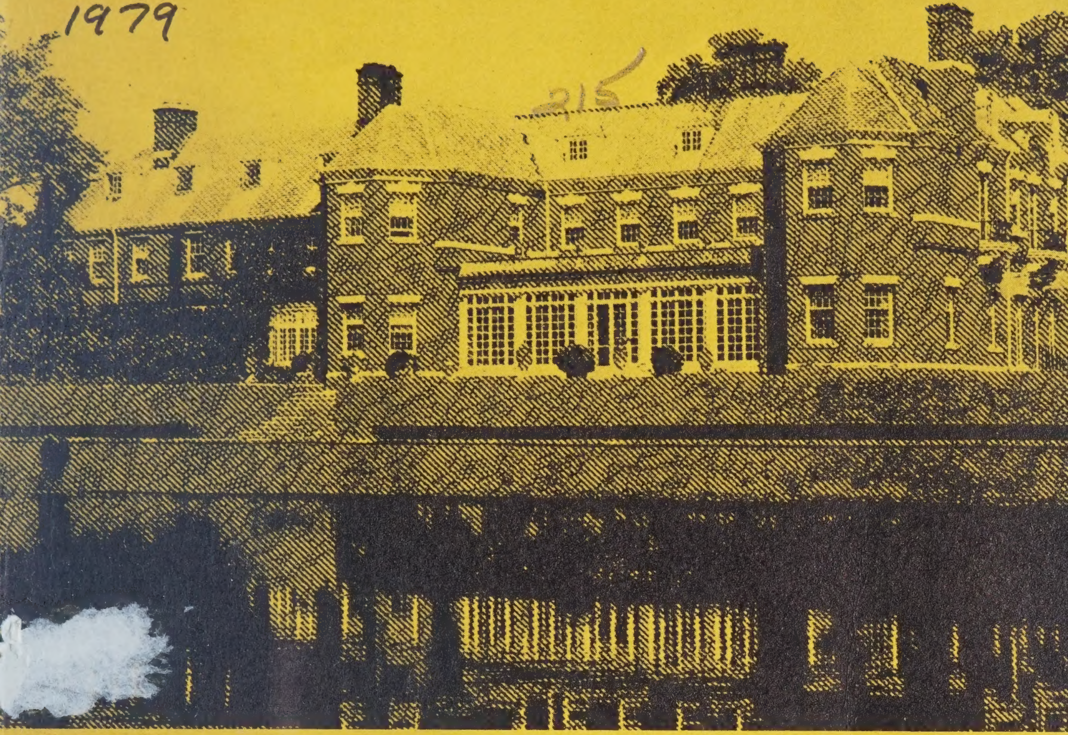
PHOTO REPRODUCTION

L161—O-1096

Digitized by the Internet Archive
in 2026 with funding from
University of Illinois Urbana-Champaign

https://archive.org/details/proceedings1979alle_0

621.3811
Al5
1979
Engin.
215



Proceedings

SEVENTEENTH ANNUAL ALLERTON CONFERENCE ON COMMUNICATION, CONTROL, AND COMPUTING

October 10-12, 1979

Allerton House, Monticello, Illinois
Sponsored by the
Department of Electrical Engineering and the
Coordinated Science Laboratory of the
University of Illinois at Urbana-Champaign

The Library of the
APR 29 1980
University of Illinois
at Urbana-Champaign

ENGINEERING LIBRARY

PROCEEDINGS

SEVENTEENTH ANNUAL ALLERTON CONFERENCE
ON COMMUNICATION, CONTROL, AND COMPUTING

J. B. Cruz, Jr.
F. P. Preparata
Conference Co-Chairmen

Conference held

October 10-12, 1979
Allerton House
Monticello, Illinois

Sponsored by

The Department of Electrical Engineering
and
The Coordinated Science Laboratory
of the
UNIVERSITY OF ILLINOIS
at
Urbana-Champaign

621.3811
Al5
1979

Engin

FOREWORD

The Seventeenth Annual Allerton Conference on Communication, Control, and Computing was held at the Allerton House on October 10-12, 1979. The Conference was sponsored by the Department of Electrical Engineering and the Coordinated Science Laboratory of the University of Illinois at Urbana-Champaign. These PROCEEDINGS contain the papers presented at the Conference.

In addition to the large number of contributed papers, there were three organized sessions of invited papers. One session on Analog Fault Diagnosis organized by R. W. Liu, one session on Reduced Order Modeling and Singular Perturbation, organized by K.-K. D. Young, and one session on Estimation and Control Applications to Aerospace Systems, organized by A. H. Haddad. We are grateful to these organizers for their time and effort on behalf of the Conference. We are also grateful to the other session chairmen and authors for their contributions to the success of the Allerton Conference.

On Wednesday evening George R. Cooper delivered the plenary address entitled "Design of Time-Frequency Coded Signals for Spread Spectrum Communications." This meeting was co-sponsored by the Joint Chapter of the IEEE Information Theory Group and was chaired by M. B. Pursley.

The support and encouragement of Professor G. W. Swenson, Jr., Head of the Department of Electrical Engineering, and Professor R. T. Chien, Director of the Coordinated Science Laboratory are appreciated and acknowledged. We thank our colleagues for their generous assistance in organizing the Conference: J. A. Abraham, D. J. Brown, B. Hajek, W. K. Jenkins, P. V. Kokotovic, M. R. Lightner, D. P. Looze, D. E. Muller, D. C. Munson, W. R. Perkins, H. V. Poor, M. B. Pursley, B. R. Rau, D. V. Sarwate, P. W. Sauer, M. Schlansker, S. Swamy, T. N. Trick, M. E. Van Valkenburg.

We are grateful to Harold Lawler for his handling of the business aspects of the Conference. Thanks are due to Rose Harris and Rosa Townsend for their skillful coordination of the Conference registration and to B. Avramovic and M. Salman for their supervision of the projectionists and visual aid equipment. Special thanks are finally due to Rose Harris for her gracious and impeccable assistance in all the organizational aspects of the Conference, from the preparation of the announcement to the editing of the Proceedings.

J. B. Cruz, Jr.
F. P. Preparata

TABLE OF CONTENTS

FOREWORD iii

I-A: ALGORITHMICS I

Chairman: J. E. Savage
Brown University

PARALLEL PERMUTATION AND SORTING ALGORITHMS 1
D. Nassimi and S. Sahni

A PARALLEL ALGORITHM FOR CONSTRUCTING MINIMUM SPANNING TREES 11
J. L. Bentley

PARALLEL ALGORITHMS AND SERIAL DATA STRUCTURES 21
J. I. Munro and E. L. Robertson

PARALLEL MATRIX AND GRAPH ALGORITHMS 27
E. Dekel, D. Nassimi, and S. Sahni

PARALLEL ALGORITHMS FOR SOLVING SYSTEMS OF NONLINEAR EQUATIONS 37
H. Mukai

BROADCASTING A SMALL NUMBER OF MESSAGES IN A SQUARE GRID GRAPH 47
F. L. Van Scoy

I-B: LARGE SCALE DYNAMIC SYSTEMS

Chairman: D. P. Looze
University of Illinois, Urbana

STRUCTURAL DECOMPOSITION AND STABILIZATION OF LARGE-SCALE CONTROL SYSTEMS 54
M. E. Sezer and D. D. Siljak

DECENTRALIZED STABILIZATION OF LARGE SCALE SYSTEMS WITH TIME DELAY 64
M. Ikeda and D. D. Siljak

CONSTRUCTION OF LYAPUNOV FUNCTIONS FOR COMPOSITE SYSTEMS BASED ON THE FREQUENCY-DOMAIN CONDITION 73
M. Saeki, M. Araki, and B. Kondo

THE PADE-HURWITZ METHOD OF MODEL REDUCTION 82
Y. Bistritz and U. Shaked

A NOTE ON ORDER-REDUCTION OF LINEAR SYSTEMS VIA PARAMETER OPTIMIZATION 92
N. Schiavoni

INFORMATION MEASURES IN DECENTRALIZED SYSTEMS 102
M. L. El-Sayed and P. S. Krishnaprasad

MODEL REDUCTION USING AN INFORMATION RATE CRITERION 110
B. S. Evans and P. S. Krishnaprasad

I-C: COMMUNICATION THEORY

Chairman: H. V. Poor
University of Illinois, Urbana

QUADRIPHASE DIRECT-SEQUENCE SPREAD-SPECTRUM MULTIPLE-ACCESS COMMUNICATION VIA FADING CHANNELS 112
D. E. Borth

OPTIMIZATION OF MULTIPLE-INTEGRATION QUANTIZED-SAMPLE RECEIVERS FOR IMPULSIVE INTERFERENCE ENVIRONMENTS 122
Y. Han and R. E. Ziemer

NONPARAMETRIC METHODS FOR DETECTION OF KNOWN SIGNALS IN DEPENDENT NOISE 133
M. DonVito and S. A. Kassam

ON THE PERFORMANCE OF A MODIFIED SIGN DETECTOR FOR M-DEPENDENT DATA 143
D. R. Halverson and G. L. Wise

DETECTION OF SIGNALS WITH UNKNOWN PHASE 152
A. H. El-Sawy

I-D: ANALOG FAULT DIAGNOSIS
Chairman: J. Morris
Office of Naval Research

Organizer: R. W. Liu
University of Notre Dame

SINGLE FAULT ANALYSIS OF ANALOG CIRCUITS WITH INACCESSIBLE NODES 162
A. A. Sakla, T. N. Trick, E. El-Masry, and B. Inn

GRAPH THEORETIC ASPECTS OF ANALOG FAULT DIAGNOSIS 164
S. D. Bedrosian and J. H. Lee

A SINGLE-FAULT DIAGNOSIS THEORY 172
C.-S. Lin and R. Liu

II-A: ADAPTIVE AND STOCHASTIC CONTROL

Chairman: P. V. Kokotovic
University of Illinois, Urbana

EXPERIMENTAL EVALUATION OF MULTIVARIABLE SELF-TUNING CONTROLLERS 179
H. N. Koivo, J. T. Tanttu, and J. Penttinen

MULTIVARIABLE TUNING REGULATORS FOR UNKNOWN SYSTEMS 188
J. Penttinin and H. N. Koivo

ON STRONG CONSISTENCY OF SOME ADAPTIVE ALGORITHMS 198
E. Fogel

SOME RESULTS ON THE IDENTIFIABILITY PROBLEM WITH APERIODIC SAMPLING 206
M. de la Sen and S. Dormido

DISCRETE SPECTRAL ESTIMATION WITH OBSERVERS 215
G. H. Hostetter

II-B: ALGORITHMICS II

Chairman: E. M. Reingold
University of Illinois, Urbana

RECORD ALLOCATION TO MINIMIZE EXPECTED SEEK DELAY 225
U. I. Gupta, D. T. Lee, J. Y-T Leung, J. W. Pruitt, and C. K. Wong

INHERENT COMPLEXITY OF RANGE QUERY PROBLEMS 231
M. L. Fredman

A MULTIKEY SEARCH PROBLEM 241
J. I. Munro

VERY FAST INFORMATION UPDATE AND RETRIEVAL USING CELLS .. 245
B. W. Weide

IMPROVING STORAGE UTILIZATION IN BALANCED TREES 255
G. N. Frederickson

ON REQUIRED SPACE FOR RANDOM SPLIT FILES 265
T. Mizoguchi

II-C: DATA COMPRESSION

Chairman: J. G. Dunham
Washington University

AN EVALUATION OF THE WYNER-ZIV RATE REGION 274
K. M. Mackenthun, Jr.

BOUNDS TO THE PERFORMANCE OF CAUSAL CODES FOR MARKOV SOURCES 284
R. K. Gilbert and D. L. Neuhoff

A UNIVERSAL CODE OF REDUCED COMPLEXITY FOR THE BINARY SOURCE 293
K. M. Mackenthun, Jr.

VARIABLE RATE ENCODING USING ADAPTIVE DELTA MODULATION .. 294
P. P. Prezas and J. L. LoCicero

A NON-UNIFORM SAMPLING APPROACH TO PREDICTIVE CODING 303
J. W. Mark and T. D. Todd

OPTIMUM FIXED-LEVEL QUANTIZATION OF THE DFT OF ACHROMATIC IMAGES 313
W. A. Pearlman

SOME PROPERTIES OF UNIFORM STEP SIZE QUANTIZERS 321
J. A. Bucklew and N. C. Gallagher, Jr.

II-D: FAULT-TOLERANT COMPUTING

Chairman: J. A. Abraham
University of Illinois, Urbana

COMPLEXITY ASPECTS OF SYSTEM DIAGNOSIS 329
P. Maestrini

ON FAULT DIAGNOSIS OF ITERATIVE LOGIC ARRAYS 339
R. Parthasarathy and S. M. Reddy

A POISSON PROCESS MODEL FOR INTERMITTENT FAULTS IN DIGITAL SYSTEMS 348
P. K. Varshney

FAULT DIAGNOSIS BY COMPRESSED TEST DATA 349
S. M. Reddy

FAULT DIAGNOSIS ALGORITHMS FOR ASYMMETRIC MODULAR ARCHITECTURES 350
G. G. L. Meyer

COMPUTER SYSTEM DIAGNOSIS ASSUMING ASYMMETRIC INVALIDATION 354
C. S. Holt and J. E. Smith

ASYMMETRIC TESTING FOR HYBRID FAULT DIAGNOSIS 364
S. Mallela and G. M. Masson

II-E: CIRCUITS AND POWER SYSTEMS

Chairman: M. E. Van Valkenburg
University of Illinois, Urbana

MODIFICATION OF ACAR-GHAUSI SENSITIVITY MEASURE WHICH INCLUDES THE NON-IDEAL EFFECTS OF THE OP. AMPS 374
J. Mulawka

NEW FORMULAS FOR A CLASS OF OPTIMAL FILTERS 381
T. S. Lim and T. N. Lee

ADDITIONAL TRANSFORMATIONS OF LIVE AND SAFE MARKED GRAPHS 387
R. Johnsonbaugh, M. C. Kao, and T. Murata

COMPUTATION OF THE EFFECT OF REAL POWER INJECTION ON THE ANGLE ACROSS TRANSMISSION LINES IN POWER NETWORKS 397
N. Narasimhamurthi

POWER SYSTEM STABILITY INCLUDING NON-TRIVIAL TRANSFER CONDUCTANCES 403
S. J. Skar

DERIVATION OF THE BRAYTON-MOSER EQUATIONS FROM A TOPOLOGICAL MIXED POTENTIAL FUNCTION 410
F. M. Massimo, H. G. Kwatny, and L. Y. Bahar

PLENARY SESSION

DESIGN OF TIME-FREQUENCY CODED SIGNALS FOR SPREAD-SPECTRUM COMMUNICATIONS 420
G. R. Cooper

III-A: REDUCED ORDER MODELING AND SINGULAR PERTURBATION

Chairman and Organizer: K-K. D. Young
Drexel University

MODAL COST ANALYSIS IN MODEL AND CONTROLLER REDUCTIONS .. 421
R. E. Skelton

SINGULAR PERTURBATIONS AND MODEL REDUCTION METHODS FOR LARGE SCALE AND DISTRIBUTED PARAMETER SYSTEMS 434
M. J. Balas

FEEDBACK DESIGN OF SINGULARLY PERTURBED SYSTEMS 449
K-K. D. Young

A FUNCTIONAL ANALYTIC APPROACH TO AUTONOMOUS LINEAR SINGULAR SINGULAR PERTURBATION PROBLEMS 455
S. L. Campbell

STABILIZATION OF TIME-VARYING SINGULARLY PERTURBED SYSTEMS BY OBSERVER-BASED SLOW-STATE FEEDBACK 464
S. H. Javid

III-B: COMBINATORIAL COMPUTING

Chairman: G. K. Manacher
University of Illinois, Chicago Circle

APPROXIMATION ALGORITHMS FOR SOME GRAPH AUGMENTATION PROBLEMS 470
G. N. Frederickson and J. Ja'Ja'

EMBEDDABILITY OF WEIGHTED GRAPHS IN k-SPACE IS STRONGLY NP-HARD 480
J. B. Saxe

FAST RECOGNITION OF BAXTER PERMUTATIONS USING SYNTACTICAL AND COMPLETE BIPARTITE COMPOSITE DAGS 490
J. M. Hart

DYNAMIC-PROGRAMMING ALGORITHMS FOR RECOGNIZING SMALL-BANDWIDTH GRAPHS IN POLYNOMIAL TIME 499
J. B. Saxe

FIRST STEPS TOWARD A NEW THEORY OF DYNAMIC PROGRAMMING 509
A. Rosenthal and P. Helman

CONTINUAL PATTERN REPLICATION 515
J. I. Munro and E. L. Robertson

III-C: NONLINEAR SYSTEMS

Chairman: A. N. Michel
Iowa State University

DECOMPOSITIONS IN NONLINEAR AUTONOMOUS SYSTEMS: I - GLOBAL 525
S. H. Mikhail

DECOMPOSITIONS IN NONLINEAR AUTONOMOUS SYSTEMS: II - LOCAL 533
S. H. Mikhail

TOPOLOGICAL CONSIDERATIONS IN SYSTEMS REALIZATION 540
R. J. Su, T. J. Tarn, and D. L. Elliott

ASYMPTOTIC STABILITY IN PARTIALLY DISSIPATIVE NONLINEAR SYSTEMS 546
R. K. Miller and A. N. Michel

ON A CLASS OF NONLINEAR ROBUSTNESS PROBLEMS 555
B. R. Barmish, L. E. Blume, and S. D. Chikte

ESTIMATION AND CONTROL OF NONLINEAR SYSTEMS HAVING CONE-BOUNDED NONLINEARITIES 562
B. M. Scherzinger and R. H. S. Kwong

DETERMINING THE FEASIBLE REGION FOR NON-STATIONARY NON-LINEAR FEEDBACK SYSTEMS 571
H. Sira-Ramirez

III-D: ESTIMATION AND FILTERING

Chairman: B. E. Hajek
University of Illinois, Urbana

IMPROVED ESTIMATORS FOR NONRANDOM WAVEFORMS IN THE PRESENCE OF NOISE 579
C. A. Pomalaza and C. D. McGillem

APPROXIMATION OF HILBERT-SPACE-VALUED STOCHASTIC INTEGRALS BY FINITE-DIMENSIONAL ONES 587
N. Berman and W. L. Root

APPROXIMATE REPRESENTATION OF BANDLIMITED SIGNALS BY ELEMENTARY FUNCTIONS 596
A. Sonnenschein and B. Dickinson

FIXED-POINT SUBOPTIMAL SMOOTHING 606
D. O. Benson, Jr.

ON THE DETERMINATION OF REGRESSION FUNCTIONS 616
G. L. Wise and N. C. Gallagher, Jr.

III-E: PARALLEL AND DISTRIBUTED COMPUTER SYSTEMS

Chairman: B. R. Rau
University of Illinois, Urbana

SCHEDULING INDEPENDENT TASKS WITH COMMUNICATIONS 624
D. H. Cornett and M. A. Franklin

NUMBERING THE NODES OF A GRAPH BY DISTRIBUTED ALGORITHMS 634
M. F. Aburdene

COMPILE TIME PREDICTION OF INTRINSIC PARALLELISM IN HIGH LEVEL PROGRAMS 640
C. Retnadhas

A STUDY OF THE AVERAGE PERFORMANCE OF A LIST TASK SCHEDULING ALGORITHM 650
A. Bashir, V. Susarala, and K. Vairavan

A SIMPLE ANALYSIS OF THE CSMA PROTOCOL 651
A. Ephremides and E. Gelenbe

IV-A: ALGORITHMICS III

Chairman: G. I. Davida
University of Wisconsin, Milwaukee

ON THE NP-COMPLETENESS OF A REAL-TIME SCHEDULING PROBLEM WITH TWO TYPES OF MACHINES 652
K. Nakajima and S. L. Hakimi

PREEMPTIVE SCHEDULING OF PERIODIC, REAL-TIME TASKS 659
J. Y-T. Leung and M. L. Merrill

ON THE STRUCTURE OF SETS IN NP AND OTHER COMPLEXITY CLASSES 661
L. H. Landweber, R. Lipton, and E. L. Robertson

AREA-TIME TRADEOFFS FOR MATRIX MULTIPLICATION AND RELATED PROBLEMS IN VLSI MODELS 670
J. E. Savage

SOME SPACE-EFFECIENT ALGORITHMS 677
J. Ja'Ja' and J. Simon

IV-B: ESTIMATION AND CONTROL APPLICATIONS TO AEROSPACE SYSTEMS

Chairman and Organizer: A. H. Haddad
National Science Foundation

APPLICATION OF ESTIMATION TECHNIQUES TO TARGET TRACKING: AN OVERVIEW OF PROBLEMS AND SOLUTIONS 685
C. B. Chang and J. A. Tabaczynski

ROBUST FILTERING AND SMOOTHING OF TRACKING DATA 695
W. S. Agee and R. H. Turner

APPROXIMATE ANALYSIS OF VARIABLE LINEAR FILTERS FOR AEROSPACE APPLICATIONS 705
R. V. Ramnath and W. M. Bowles

AIRCRAFT TRAJECTORY OPTIMIZATION USING SINGULAR PERTURBATION METHODS 712
R. K. Mehra and R. S. Washburn, Jr.

OPTIMIZED GUIDANCE AND PROPULSION CONTROL FOR RAMJET PROPELLED VEHICLES BASED ON SINGULAR PERTURBATION METHODS 725
A. Calise and F. Goldstein

APPLICATION OF OPTIMAL CONSTANT OUTPUT FEEDBACK TO FLIGHT CONTROL SYSTEM DESIGN 735
E. Y. Shapiro and D. A. Fredericks

IV-C: CONTROL SYSTEMS I

Chairman: H. K. Khalil
Michigan State University

OPTIMAL SENSITIVITY AND FEEDBACK: WEIGHTED SEMINORMS, APPROXIMATE INVERSES, AND PLANT INVARIANT SCHEMES 744
G. Zames

LOOP CLOSURES AND THE INDUCED EXTERIOR MAP 753
V. Seshadri and M. K. Sain

ABSTRACT BILINEAR SYSTEMS: THE FORWARD SHIFT APPROACH .. 762
A. E. Frazho

TEAM-OPTIMAL CLOSED-LOOP STACKELBERG STRATEGIES IN LINEAR-QUADRATIC DIFFERENTIAL GAMES 771
T. Basar and G. J. Olsder

DISTURBANCE ISOLATION VIA HIGH GAIN FEEDBACK 779
H. K. Khalil

DESIGN OF OPTIMAL TRACKING DISCRETE CONTROL SYSTEM 787
Y. Bar-Ness and J. Feinstein

MICROPROCESSOR BASED COMPENSATION FOR NONLINEAR FRICTIONS 798
D. H. Chyung

IV-D: COMMUNICATIONS

Chairman: D. E. Borth
Georgia Institute of Technology

PERFORMANCE BOUNDS FOR CONVOLUTIONAL CODES WITH MISMATCHED DIGITAL VITERBI DECODERS 802
J. G. Dunham and K-H. Tzou

ON THE PERFORMANCE OF CERTAIN BLOCK ERROR-CORRECTING CODES IN BINARY CHANNELS 810
Z. A. Muscati and S. G. S. Shiva

MATHEMATICAL MODEL FOR A NEW COMMUNICATION SYSTEM ENABLING SIMULTANEOUS TRANSMISSION OF SPEECH AND DATA ... 819
M. Nagi El-Yousfi and M. Moustafa Rateb

BOUNDS ON MESSAGE START INFORMATION FOR MESSAGES SUBJECT TO QUEUEING DELAY 820
K. M. Mackenthun, Jr.

QUANTIZED NOISE PROTECTION AND MULTICODING FOR AN OPTIMUM BINARY ARQ COMMUNICATION 830
M-E. Moustafa Rateb and M. Nagi El-Yousfi

ON THE PROPOSED FEDERAL STANDARD 1033 FOR TELECOMMUNICATIONS 840
E. D. Eyman

V-A: DETECTION, ESTIMATION, AND FILTERING

Chairman: K. M. Mackenthun, Jr.
University of South Carolina

GENERALIZATIONS OF THE SEQUENTIAL SIGN DETECTOR 848
C. C. Lee and J. B. Thomas

SUBOPTIMAL DETECTION SCHEMES WITH COUNTING POINT PROCESS OBSERVATIONS 858
A. A. Lazar and S. C. Schwartz

DETECTION OF SIGNALS WITH RANDOM PHASE IN SEA CLUTTER ... 866
A. H. El-Sawy

GLOBAL IDENTIFICATION OF CONTINUOUS-TIME SYSTEMS WITH UNKNOWN NOISE COVARIANCE 876
J. K. Tugnait

MEDIAN FILTERING 886
V. K. Aatre, E. Ataman, and K. M. Wong

EDGE-ADAPTIVE PREDICTORS FOR DPCM CODING OF PICTURES 896
P. Cohen and A. Valette

V-B: CONTROL SYSTEMS II

Chairman: A. Thowsen
Georgia Institute of Technology

SOME EXPERIMENTAL, ANALYTICAL AND COMPUTATIONAL RESULTS CONCERNING MAXIMAL VERTICAL JUMPS 904
W. S. Levine, F. E. Zajac, M. R. Zomlefer, and M. R. Belzer

CONTROL OF MULTIFUNCTIONAL ABOVE-ELBOW AND SHOULDER-DISARTICULATION PROSTHESES BY SINGLE-SITE EMG TEMPORAL SIGNATURE IDENTIFICATION 914
D. Graupe, J. Salahi, and J. Grosspietsch

MODELING HUMAN PERFORMANCE IN AN ANTI-AIRCRAFT TRACKING TASK WITH DELAYED MEASUREMENTS 919
K-C. Wei and M. Vikmanis

CONTINUOUS SIMULATION OF TRANSIENT AND TIME VARYING CONDITIONS IN QUEUEING NETWORKS 920
K. G. Wang and S. S. L. Chang

A RESULT IN THE CONTROL OF SYSTEMS GOVERNED BY LINEAR DIFFERENTIAL-DIFFERENCE EQUATIONS 930
A. Thowsen

ROBUST ITERATIVE ALGORITHMS 937
A. J. David and G. G. L. Meyer

V-C: ALGORITHMICS IV

Chairman: D. E. Muller
University of Illinois, Urbana

SORTING IN MAGNETIC BUBBLE MEMORY 940
K. M. Chung, F. Luccio, and C. K. Wong

FURTHER RESULTS ON NEAR-OPTIMAL SORTING 949
G. K. Manacher

THE UP-DOWN ALGORITHM FOR TWO-DIMENSIONAL BIN PACKING ... 961
B. S. Baker, D. J. Brown, and H. P. Katseff

AN ALGORITHM FOR THE HITCHCOCK TRANSFORMATION PROBLEM ... 962
D. P. Bertsekas

ON THE COMPLEXITY OF STEINER-LIKE PROBLEMS 969
P. M. Camerini, G. Galbiati, and F. Maffioli

REAL AND COMPLEX TENSORS OF NEAR MAXIMAL RANK 978
C. M. Fiduccia

V-D: DIGITAL SYSTEMS AND APPLICATIONS

Chairman: W. K. Jenkins
University of Illinois, Urbana

MULTIPROCESSING COMPUTER ORGANIZATION WITH MULTIPLE READ SINGLE WRITE MEMORY 987
S. S. L. Chang

REVOLUTIONARY DESIGNS FOR ASYNCHRONOUS SEQUENTIAL CIRCUITS USING LSI 996
G. K. Maki and J. L. Keel

FFT ALGORITHMS FOR SIMD MACHINES 1006
L. J. Siegel, P. T. Mueller, Jr., and H. J. Siegel

CAPACITY CALCULATION OF COMPOSITE CONCENTRATORS 1016
S. Nakamura and G. M. Masson

A METHOD TO DESIGN RECURSIVE DIGITAL FILTERS 1026
J. Rivero Laguna

PARALLEL PERMUTATION AND SORTING ALGORITHMS*

DAVID NASSIMI+ AND SARTAJ SAHNI
University of Minnesota
Minneapolis, Minnesota 55455

Abstract: We obtain O(k log N) algorithms to permute and sort N data items on cube and perfect shuffle computers with $N^{1+1/k}$ processing elements, $1 \leq k \leq \log N$. These algorithms directly lead to a generalized-connection-network construction having O(k log N) delay and O(k $N^{1+1/k}$ log N) contact pairs. This network has the advantage that the switches can be set in O(k log N) time by either a cube or perfect shuffle computer with $N^{1+1/k}$ processing elements.

Key words and phrases: SIMD machines, parallel algorithms, permutation, sorting, cube computer, perfect shuffle computer, generalized connection network.

1. Introduction

An SIMD (Single Instruction Stream, Multiple Data Stream) type computer system is comprised of some number of processing elements (PEs), each having some local memory. We shall assume that the PEs are indexed 0 through p-1 and shall refer to the ith PE as PE(i). The PEs are synchronized and operate under the control of a single instruction stream. An enable/disable mask may be used to select a subset of PEs that will perform the instruction to be executed at any given time. All enabled PEs perform the same instruction. Three SIMD architectures are relevent to our discussion:

i) Shared Memory Model (SMM)

In this model there is a common memory available to all PEs. This is in addition to the local memory available to each PE. Data may be transmitted from PE(i) to PE(j) by simply having PE(i) write the data into the common memory and then letting PE(j) read it. It thus takes O(1) time to transmit data from one PE to another in this model. Two PEs are not permitted to write into the same word of common memory simultaneously. PEs may or may not be allowed to simultaneously read the same word of common memory. Algorithms that require two or more PEs to read the same common memory word simultaneously will be said to have memory fetch conflicts.

ii) Cube Connected Computers (CCC)

Assume that $p = 2^q$ and let $i_{q-1} \ldots i_0$ be the binary representation of i for $i \in [0, p-1]$. Let $i^{(b)}$ be the number whose binary representation is $i_{q-1} \ldots i_{b+1} \bar{i}_b i_{b-1} \ldots i_0$ where $\bar{i}_b$ is the complement of i_b and $0 \leq b < q$. In the cube model, PE(i) is connected to PE($i^{(b)}$), $0 \leq b < q$. Data can be transmitted from one PE to another only via the interconnection pattern.

iii) Perfect Shuffle Computer (PSC)

Let p, q, i and $i^{(b)}$ be as in the cube model. In the perfect shuffle model

---

*This research was supported in part by NSF grant MCS 78-15455 and represents a portion of the first author's doctoral dissertation.

+Present address: Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60201.

PE(i) is connected to PE($i^{(0)}$), PE ($i_{q-2}i_{q-3}\cdots i_0 i_{q-1}$) and PE ($i_0 i_{q-1} i_{q-2}\cdots i_1$). These three connections will respectively be called *exchange*, *shuffle* and *unshuffle*. Once again, data transmission from PE to PE is possible only via the connection scheme.

It should be noted that the CCC model requires log p (all logarithms in this paper are base 2) connections per PE and the PSC model requires only three connections per PE. The SMM requires a large amount of PE to memory connections to permit simultaneous memory access by several PEs. Each of these models has received much attention in the literature (see for example [1,3,4,5,7,8,10,11,13,17,19,20, and 21]). In this paper we are concerned with the development of efficient algorithms to sort and permute data on a CCC and a PSC. For both problems we shall assume that there are $N=2^n$ records. Initially, record i is in PE(i), $0 \le i < N$. Each record i has a field A(i). In the sorting problem the N records are to be rearranged into nondecreasing order of A(i), one record to a PE. For the permutation problem, $A(i) \in [0, N-1]$, $0 \le i < N$, and record i is to be relocated to PE(A(i)), $0 \le i < N$. N records can be permuted in O(1) time using the SMM and N PEs. PE(i) first writes its record into location A(i) of the common memory (assuming the common memory is large enough to hold N records) and then reads back the record in location i. Clearly, this algorithm has no read/write conflicts. N records can be sorted on an N-PE SMM in $O(\log^2 N)$ time, using Batcher's algorithm [2]. Hirschberg [8] and Preparata [17] have developed sorthing algorithms for SMM for the case when $N^{1+1/k}$ PEs are available. Their algorithms run in O(k log N) time. Hirschberg's algorithm has memory fetch conflicts while Preparata's does not. Thompson and Kung [22] and Nassimi and Sahni [14] show how to sort $N=n^2$ data on an Illiac IV-like n x n Mesh-Connected Computer (MCC) in O(n) time. Both these papers also generalize their results to higher dimensional MCCs. For the permutation problem on MCCs, Nassimi and Sahni [15] have developed an algorithm that is optimal. Their algorithm, however, can be used on only a special class of permutations.

It is well known [19] that N records can be sorted in $O(\log^2 N)$ time on both an N-PE CCC and an N-PE PSC. Algorithms to perform certain permutations in less time on these models can be found in [10, 11, and 19]. In section 2, we develop an algorithm to perform arbitrary permutations in O(k log N) time on a CCC and a PSC with $N^{1+1/k}$ PEs. In section 3, we develop an algorithm for the sorting problem. This algorithm has the same asymptotic complexity as the permutation algorithm of section 2.

2. Permutations

Let G(i), $0 \le i < N = 2^n$, be N records. G(i) is initially located in PE(i). Let A(i) be a field in record G(i) such that A(0),..., A(N-1) defines a permutation of (0, 1,..., N-1). The records are to be permuted so that following the permutation, G(i) is in PE(A(i)), $0 \le i < N$. We shall assume that this permutation is to be performed on a CCC or a PSC having $N^{1+1/k}$ PEs, where k = n/m for some integer m, $1 \le m \le n$. For purposes of discussion, we shall view these $N^{1+1/k}=2^{n+m}$ PEs as arranged in a $2^m \times 2^n$ array. The PEs are indexed in row-major order with the result that record G(i) is initially located in the PE in row 0 and column i, $0 \le i < N$.

The permutation algorithm to be described is essentially a parallel version of MSD (most significant digit) radix sort (see Knuth [9]). The radix used is 2^m. Using this radix, the $\lceil n/m \rceil$ digits of A(i) are obtained as follows. First, the binary representation of A(i) is obtained. The m most-significant bits yield the most-significant digit, the next m bits give the next digit and so on. For example, if m = 2 and n = 3, then all numbers in the range [0,7] are represented as d_1d_0 with radix 2^m=4. The representations are 00, 01, 10, 11, 20, 21, 30, and 31. An example will illustrate how the permutation algorithm proceeds. Let us assume that m = 2 and n = 3. Hence, 8 records are to be permuted using 32 PEs. Figure 2.1.a represents a possible initial configuration. Each square represents a PE and the number within a square is the value of A(i). The earlier example gives the d_1 and d_0 values of each A(i). Since there are two digits (using radix 2^m) in the representation of the A(i)s, the permutation algorithm will go through two phases. In the first, the records will be ordered with respect to d_1 and in the second, with respect to d_0. In phase 1, the record in PE(i) is routed to the PE in column i and row D1(i), where D1(i) is the d_1 value of A(i). Figure 2.1.b shows the configuration following this routing. Next, the records are ranked. The rank of a record in PE(i) is equal to the number of records located in PEs with index less than i (recall that PEs are indexed in row major order). Let R(i) be the rank of the record in column i (we shall always have only one record in each column of PEs). For Figure 2.1.b we have R(0:7) = (4,2, 0,5,1,6,7,3). Having determined the ranks, the records are routed to columns with index equal to the record rank (note that columns are numbered 0 through N-1). During this routing, records do not change rows. This results in the configuration of Figure 2.1.c. The routing carried out will be referred to as concentration. This completes phase 1.

Following phase 1, records are ordered (by columns) with respect to d_1. It remains to perform the ordering with respect to d_0. This is done independently for groups of records having the same d_1 value. For the example, there are four such groups. Each group consists of two columns, 2i and 2i + 1. The procedure for this ordering is the same as that for phase 1. Records are routed to row d_0 without changing the column (Figure 2.1.d). Records are then ranked within each group and routed to the appropriate column in that group without changing rows (Figure 2.1.e). This completes phase 2. At the end of this phase, record G(i) is located in column A(i). (In general, $\lceil n/m \rceil$ phases, as described above, will be needed to achieve this condition.) To obtain the desired final record distribution, the records are routed to row 0 without changing the column of any record (Figure 2.1.f).

5	2	1	4	0	7	6	3

(a)

		1		0			
	2						3
5			4				
					7	6	

(b)

1	0						
		2	3				
				5	4		
						7	6

(c)

	0	2			4		6
1			3	5		7	

(d)

0		2		4		6	
	1		3		5		7

(e)

0	1	2	3	4	5	6	7

(f)

Figure 2.1 Permutation of 8 elements on 4 x 8 PEs

To summarize, our permutation algorithm consists of $\lceil n/m \rceil$ route - rank - concentrate phases, followed by a routing phase in which all records are routed to row 0.

Permutation Algorithm For A CCC

We now implement the permutation procedure described above. The algorithm for a CCC is specified first. We shall see, later, that the algorithm for a PSC can be easily obtained from that for a CCC. In specifying the algorithms, we make use of the following notation and assumptions:

1) Each PE has three registers R, S and T. R(i), S(i) and T(i) refer to the corresponding registers in PE(i). In addition, each PE has enough memory to hold one record G (this includes the field A).

2) We shall use the special symbol null. The only requirement is that it be possible to distinquish between null and a record (a one bit tag could be used).

3) i_b will denote bit b in the binary representation of i. And, $(i)_{j:\ell}$, $j \geq \ell$, will denote the integer with the binary representation $i_j i_{j-1} \ldots i_{\ell+1} i_\ell$.

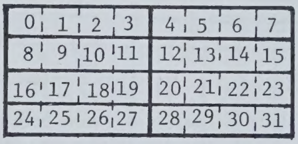
4) PE selectivity can be done using a mask. The mask is specified in parenthesis following the statement. Some examples of masks are:

i) $(i_b=1)$: this enables all PEs for which the binary representation of the PE index has bit b equal to 1.

ii) (A(i) = null) : this enables all PEs with A(i) = null.

When no mask is specified, all PEs are enabled. Instructions are executed only on enabled PEs.

The complexity of an algorithm will be measured in terms of the PE time needed and the number of unit-routes. We shall now develop the algorithms corresponding to the functions rank and concentrate used in our earlier discussion. From these it will be easy to obtain the complete permutation algorithm. Both RANK and CONCENTRATE operate on groups of 2^k consecutive PEs, $0 \leq k \leq n$. The PE grouping is the natural one with PEs $\{2^k i,\ldots,2^k(i+1)-1\}$, $0 \leq i < 2^{n-k}$, of any row forming a group. Figure 2.2 shows the PE groups when n = 3, m = 2, and k = 2. There are eight groups, each consisting of four PEs. When k = n, each row makes up one group. Note that the PEs in the same group of size 2^k may differ only in the least significant k bits of their indices.



0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31

Figure 2.2 8 groups of 4 PEs each

The rank of a record in a 2^k PE group is the number of records preceding it in that group. Procedure RANK determines this quantity for each record in every group. The working of RANK is best described recursively. Divide a 2^k group into two halves, each half containing 2^{k-1} consecutive PEs. Let R(i) be the rank of the record (if any) in PE(i) (i.e. rank within the 2^{k-1} group). Let S(i) be the total number of records in the 2^{k-1} group containing PE(i). Then, the rank of a record in a 2^k group is R(i) if $i_{k-1}=0$ (note that $i_{k-1} = 0$ for the left 2^{k-1} group of a 2^k group) and $R(i) + S(i^{(k-1)})$ if $i_{k-1}=1$. Unfolding the recursion yields the iterative procedure RANK. It is easy to see that RANK uses $O(k)$ PE time and exactly k *unit-routes*. (In a unit-route, each enabled PE can transmit its data to another PE directly connected to it.)

```
procedure RANK (k)
  //Rank records in groups of $2^k$ consecutive PEs//
    global A, R, S
  //initialize for single PE groups//
    $R(i) := 0$
    $S(i) := 1, (A(i) \neq$ null$)$
    $S(i) := 0, (A(i) =$ null$)$
    for b := 0 to k-1 do
       $T(i^{(b)}) \leftarrow S(i)$
       $R(i) := R(i) + T(i), (i_b = 1)$
       $S(i) := S(i) + T(i)$
    end
end RANK
```

Procedure CONCENTRATE concentrates records within a 2^k PE group. Recall that during concentration, records are moved to consecutive PEs. Let R(i) be such that the record (if any) in PE(i) is to be moved to the R(i)th PE in the 2^k group. (The R(i) used here differs from the R(i) obtained by RANK by an additive constant.) Concentration is carried out by first moving all records in the group to PEs such that the PE index and R(i) agree in bit 0. The next routing assures that PE indices and R(i) agree in bits 0 and 1; and so on until records have been routed to the correct PEs.

```
procedure CONCENTRATE (k)
  //route records to appropriate PEs as determined by R//
    global G, R  //A is a field in G; G denotes the record//
    for b := 0 to k-1 do
       $<G(i^{(b)}), R(i^{(b)})> \leftrightarrow <G(i), R(i)>$ , $(A(i) \neq$ null and $R(i)_b \neq i_b)$
    end
end CONCENTRATE
```

A proof of the correctness of CONCENTRATE can be found in [23].

Using procedures RANK and CONCENTRATE and our earlier discussion of how permutations are to be performed, one can obtain a complete permutation algorithm with complexity $O(k \log N)$. The number of PEs used by the algorithm is $N^{1+1/k}$. The CCC algorithm is easily simulated by a PSC with $N^{1+1/k}$ PEs. The corresponding simulation also has complexity $O(k \log N)$. See [23] for details.

3. Sorting

$N = 2^n$ elements can be sorted in $O(k \log N)$ time on a CCC or PSC having $N^{1+1/k}$ PEs. The algorithm is quite similar to that developed by Preparata [17] for the shared memory model. Preparata's algorithm consists of three distinct steps:

i) *Counting*: Divide the N elements into some number j of subsets. For each element A_i in subset ℓ determine the number, C_{ip}, of elements in subset p that are:

a) no greater than A_i, $p < \ell$
b) no greater than A_i and to the left of A_i, $p = \ell$
c) less than A_i, $p > \ell$.

ii) Ranking: The rank, R_i of the element A_i is its final position in the sorted sequence. This is just the sum of the counts for that element. That is, $R_i = \sum_p C_{ip}$.

iii) Routing: Each element is routed to the PE corresponding to its rank.

We shall show that these three steps can be implemented on a CCC or PSC to obtain a sorting algorithm with the same asymptotic complexity as that of Preparata's algorithm. We shall describe our implementation explicitly for a CCC only. The PSC algorithm may be obtained by simulating the CCC algorithm (see [23]). We assume that 2^{n+m}, $1 \leq m \leq n$, PEs are available. These PEs may be viewed as arranged in a $2^m \times 2^n$ array. The elements to be sorted are A(0:N-1), and initially the i th element is in PE(i). For the remaining PEs, A = null.

Before discussing the sorting algorithm (procedure SORT), we introduce some terminology. By a 2^r - block of PEs we shall mean a group of 2^r consecutive PEs. PE blocks will be constructed in the same way as in section 2. PE(i) is a diagonal PE under 2^r - blocking iff

$$\lfloor i/2^n \rfloor = \lfloor i/2^r \rfloor \bmod 2^m \quad (3.1)$$

Figure 3.1 shows the diagonal PEs when r = 1, m = 2 and n = 4.

D	D							D	D						
		D	D							D	D				
				D	D							D	D		
						D	D							D	D

Figure 3.1 Diagonal PEs under 2^r-blocking, r = 1.

A left PE is one for which

$$\lfloor i/2^n \rfloor > \lfloor i/2^r \rfloor \bmod 2^m \quad (3.2)$$

and a right PE is one for which

$$\lfloor i/2^n \rfloor < \lfloor i/2^r \rfloor \bmod 2^m \quad (3.3)$$

Let DIAG (r, i), LEFT (r, i), and RIGHT (r, i) respectively be Boolean functions that are true iff (3.1), (3.2), and (3.3) hold for r and i.

The initial call to SORT is SORT (n). The procedure begins by dividing the 2^k elements into 2^m groups of 2^{k-m} consecutive elements. These are sorted by a recursive invocation of the algorithm (line 1). Following this sort, there will be exactly one element per column of PEs. This element is replicated over the column in register S (lines 3-6). If r is as defined in line 2, then at this time, each row of PEs contain sorted sequences of length 2^r (these are in the S registers). The sequences in the diagonal PEs under 2^r - blocking are next replicated over rows (each row consists of 2^k consecutive PEs). This is done in lines 7-12. Now, each 2^r - block of PEs contains a sorted sequence in its S registers and another in its T registers. For a 2^r - block in the ith group and jth row, the S sequence is from group i and the T sequence from group j. Procedure COUNT (specified in [23])determines the count R(i) corresponding to each element S(i), as discussed earlier in our informal presentation of counting.

The rank of an element can now be determined by summing up the count values for that element. The count values in each column of PEs need to be summed (lines 14-16).

The final step is routing to the appropriate PEs. This is done by first routing the elements in each diagonal PE (note that A was diagonalized in line 9) to the PE on the same row but in the correct column (lines 17-19). Each 2^k - block contains a diagonal 2^r - block. So, the elements from the 2^r - block are spread out within the 2^k - block (while preserving their relative order). This is just the inverse of concentration and so the results of section 2 establish the correctness of lines 17-19. Finally, in lines 21-23, the elements are routed up the columns to the PEs in the first row (the code of lines 21-23 actually replicates the element over the whole column).

Procedure COUNT considers each 2^r - block of PEs in parallel. Each such block contains a sorted S and T sequence. For each element S(i) in the block, the procedure is to find the corresponding count R(i). If the 2^r - block is a diagonal block, then the S and T sequences are identical. The count, R, for the ith S in a 2^r diagonal block is i. If the 2^r - block is a left block, then the count for the ith S is the number of T values in the T sequence that are less than it. When the 2^r - block is a right block, then the count for the ith S is the number of T values not greater than it. The count, R, is determined by merging the S and T sequences together (a stable merge is used). Figure 3.2 shows an S and T sequence. The arrows from row 1 to row 2 show where an S or T element gets positioned following the merge. It should be easy to see that if the ith S value is located in position j following the merge, then its R value is j - i. The third row of Figure 3.2 gives the R values for the S sequence. The implementation of COUNT is straightforward, and can be found in [23]. COUNT(r) uses exactly 3r unit-routes.

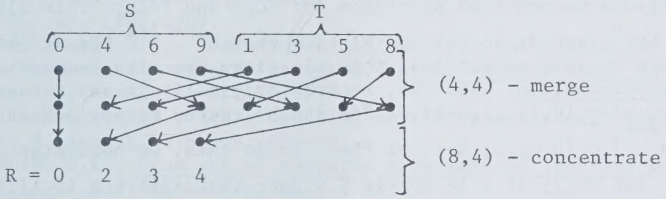


Figure 3.2 Sketch of COUNT

The total number of unit-routes used by SORT(n) is easily seen to be O(k log N), where $N = 2^n$ is the number of elements to be sorted, 2^{n+m} is the number of PEs, and $k = \frac{n}{m}$.

```
line   procedure SORT (k)
         //Sort the records in each group of $2^k$ consecutive columns//
         //in a $2^m \times 2^n$ CCC. Each column contains exactly one//
         //element. This is in register A of the appropriate PE//
         global A, R, S, T, m, n
 1       if k > m then call SORT (k - m)
 2       r := max (k - m, 0)// $2^r$ is the block size//
         //counting//
 3       S(i) := A(i)
 4       for b := n to n + m - 1 do //copy over columns//
 5           $S(i^{(b)}) \leftarrow S(i), (S(i) \neq$ null)
 6       end
```

```
        T(i) := S(i), (DIAG(r, i))
        T(i) := null, (not DIAG(r, i))
        A(i) := T(i) //A is diagonalized//
        for b := r to k - 1 do //copy over rows//
            $T(i^{(b)}) \leftarrow T(i)$, $(T(i) \neq$ null)
        end
        call COUNT (r)
        //ranking//
        for b := n to n + m - 1 do //column sum//
            $R(i) \leftarrow R(i) + R(i^{(b)})$
        end
        //routing//
        //route each diagonal block to columns given by R//
        for b := k - 1 down to 0 do
            $<A(i^{(b)}), R(i^{(b)})> \leftrightarrow <A(i), R(i)>$, $(A(i) \neq$ null and $R(i)_b \neq i_b)$
        end
        if k < n then return
        for b := n to n + m - 1 do //route to first row//
            $A(i^{(b)}) \leftarrow A(i)$, $(A(i) \neq$ null)
        end
    end SORT
```

4. Concluding Remarks

We have developed a permutation algorithm for CCCs and PSCs. This algorithm permutes $N=2^n$ records in $O(k \log N)$ time when $N^{1 + 1/k}$ PEs are available. It should be pointed out that this algorithm can also be used to perform any subpermutation. That is, records originally in any subset $S=\{s_0,s_1,\dots,s_{j-1}\} \subseteq \{0,1,\dots,N-1\}$ of PEs can be routed to any subset $D=\{d_0,d_1,\dots,d_{j-1}\} \subseteq \{0,.,\dots,N-1\}$ of PEs. To do this, we need simply to set $A(i)=$null for PE(i) if i is not in S. Note that $A(i) \in D$ if $A(i) \neq$ null.

Our sorting algorithm has the same asymptotic complexity as the permutation algorithm. However, it is slower than the permutation algorithm by a constant factor. The sorting algorithm also has the same asymptotic complexity as that developed by Preparata [17] for the shared memory model. Procedure SORT is easily adapted to sort records on a subset $S=\{s_0,s_1,\dots,s_{j-1}\} \subseteq \{0,1,\dots,2^{n+m}-1\}$ of PEs, $|S| =j \leq 2^n=N$. If $A(i)$, $i \in S$, are the sort keys, then following the sort we should have $A(s_i) \leq A(s_{i+1}), 0 \leq i < j-1$. This can be done as follows:

i) Rank the As to get $R(s_i)=i$, $0 \leq i < j$.

ii) Attach the originating address s_i to the record in $PE(s_i)$. Denote this by $OA(s_i)$.

iii) Concentrate the records in S according to R (this also concentrates the OAs). Following this the records to be sorted are in PE(0),...,PE(j-1).

iv) Set $A(i)=\infty$ for $j \leq i < N$.

v) Use SORT to sort the j records. During the sort the OA values are not shifted.

vi) Route the record in PE(i) to PE(OA(i)), $0 \leq i < j$. This is just the reverse of a concentrate.

Preparata and Vuillemin [24] have proposed a new interconnection network. The resulting parallel computer requires less chip area than one using either the cube connected interconnection or the perfect shuffle interconnection. The number of connections per PE is 3 (the same as in a PSC). The permutation and sorting algorithms developed in this paper can be adapted to run on the computer model of [24] in the same asymptotic time as on a CCC or PSC.

The sorting algorithm of this paper directly leads to a new generalized connection network [23]. This network has $0(k \log N)$ delay and uses $0(kN^{1+1/k} \log N)$ contact pairs. While it is slower than that of [21] and also uses more contact pairs, it can be set-up in $0(k \log N)$ time with a CCC or PSC with $N^{1+1/k}$ PEs. The fastest known set-up algorithms for the network of [21] take $0(k \log^3 N)$ time on a CCC or PSC with $N^{1+1/k}$ PEs (see [16]).

References

1. Arjomandi, E., "A study of parallelism in graph theory," Ph.D. thesis, Computer Science, University of Toronto, Dec., 1975.
2. Batcher, K.E., "Sorting networks and their application," *Proc. AFIPS 1968 SJCC,* Vol. 32, AFIPS press, Montvale, N.J., pp. 307-314.
3. Brent, R.P., "The parallel evaluation of general arithmetic expressions," *JACM*, Vol. 21, No. 2, April 1974, pp. 201-206.
4. Csanky, L. "Fast parallel matrix inversion algorithms, *"Proc. 6th IEEE Symp. on Found. Comput. Sc.*, Oct. 1975, pp. 11-12.
5. Eckstein, D., "Parallel graph processing using depth-first search and breadth-first search," Ph.D. thesis, University of Iowa, 1977, 180 pages.
6. Flynn, M.J., "Very high speed computing systems," *Proc. IEEE,* 54, Dec. 1966, pp. 1901-1909.
7. Hirschberg, D.S., "Parallel algorithms for the transitive closure and the connected component problems," *Proc. 8th ACM symp. on Theo. of Comput.*, May 1976, pp. 55-57.
8. Hirschberg, D.S., "Fast parallel sorting algorithms," *CACM*, Vol. 21, No. 8, August 1978, pp. 657-661.
9. Knuth, D.E., *"The art of computer programming: sorting and searching, Vol. 3,"* Addison-Wesley, Mass., 1973.
10. Lang, T., "Interconnections between processors and memory modules using the shuffle-exchange network," *IEEE Trans. on Computers*, C-25, No. 5, May 1976, pp. 496-503.
11. Lang, T. and Stone, H., "A shuffle exchange network with simplified control," *IEEE Trans. on Computers*, C-25, No. 1, Jan. 1976, pp. 55-65.
12. Muller, D.E. and Preparata, F.P., "Bounds to complexities of networks for sorting and for switching," *JACM*, Vol. 22, No. 2, April 1975, pp. 195-201.
13. Munro, I. and Paterson, M., "Optimal algorithms for parallel polynomial evaluation," *JCSS*, Vol. 7, 1973, pp. 189-198.
14. Nassimi, D. and Sahni, S., "Bitonic sort on a mesh-connected parallel computer," *IEEE Trans. on Computers*, C-28, No. 1, Jan. 1979, pp. 2-7.
15. Nassimi, D. and Sahni, S., "An optimal routing algorithm for mesh-connected parallel computers," *JACM*, Vol. 27, No. 1, Jan. 1980, to appear.
16. Nassimi, D. and Sahni, S., "Parallel algorithms to set-up Benes permutation network," University of Minnesota, Technical Report #79-19, 1979.
17. Preparata, F.P., "New parallel-sorting schemes," *IEEE Trans. on Computers*, C-27, No. 7, July 1978, pp. 669-673.
18. Savage, C., "Parallel algorithms for graph theoretic problems," Ph.D. thesis, University of Illinois, Urbana, Aug. 1978.

19. Stone, H., "Parallel processing with the perfect shuffle," IEEE Trans on Computers, C-20, 1971, pp. 153-161.
20. Swanson, R.C., "Interconnections for parallel memories to unscramble p-ordered vectors," IEEE Trans. on Computers, C-23, No. 11, 1974, pp. 1105-1115.
21. Thompson, C.D., "Generalized connection networks for parallel processor interconnection," IEEE Trans. on Computers, C-27, No. 12, Dec. 1978, pp. 1119-1125.
22. Thompson, C.D. and Kung, H.T., "Sorting on a mesh-connected parallel computer," CACM, Vol. 20, No. 4, 1977, pp. 263-271.
23. Nassimi, D. and Sahni, S., "Parallel permutation and sorting algorithms and new generalized connection network," University of Minnesota, Technical Report #79-8, 1979.
24. Preparata, F. and Vuillemin, J., "The cube-connected cycles: A versatile network for parallel computation, 20th Annual Symp. on Foundations of Computer Science, Puerto Rico, 1979.

A PARALLEL ALGORITHM FOR CONSTRUCTING MINIMUM SPANNING TREES[1]

Jon Louis Bentley
Departments of Computer Science and Mathematics
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

Abstract -- The construction of minimum spanning trees (MSTs) of weighted graphs is a problem that arises in many applications. In this paper we will study a new parallel algorithm that constructs the MST of an N-node graph in time proportional to N lg N, on an N/(lg N)-processor computing system. The primary theoretical contribution of this paper is the new algorithm, which is an improvement over Sollin's parallel MST algorithm in several ways. On a more practical level, this algorithm is appropriate for implementation in VLSI technology.

1. Introduction

The minimum spanning tree (MST) of a weighted graph is defined as that subset of edges of the graph that connects all nodes with minimum total edge weight. The problem of computing MSTs arises in many applications. In a communications system, for example, we might wish to link together a set of cities using minimum wire length; if only wires from one city to another are allowed, this calls precisely for finding the minimum spanning tree. (Indeed, this method is used for billing purposes by the Bell System!) The problem of computing MSTs also arises in such diverse applications areas as data analysis, operations research, and transportation networks. In this paper we will investigate a new method for rapidly computing MSTs on a parallel computer.

Much previous work has been devoted to the minimum spanning tree problem. In its most general setting, the problem is defined on graphs; Cheriton and Tarjan [1976] provide a thorough treatment of algorithms for the general problem. A

---

[1] This research was supported in part by the Defense Advanced Research Projects Agency under Contract F33615-78-C-1551 (monitored by the Air Force Office of Scientific Research) and in part by the Office of Naval Research under Contract N00014-76-C-0370.

special case of the MST problem (and one of great practical interest) is when the input graph is the complete graph induced by a set of points in Euclidean k-space. (That is, the vertices are the points in the set, and the edge weight between two vertices is their distance in space.) Shamos [1978] gives an optimal worst-case algorithm for computing the MST of a planar point set, and Yao [1977] describes the best-known worst-case algorithm for point sets in k-space. Fast expected-time algorithms are considered by Bentley, Weide and Yao [1978].

In this paper we will study a parallel algorithm for computing MSTs of graphs in which the edge weights are implicitly given by a distance function that returns the (nonnegative) distance between two nodes. The most common example of such graphs is the Euclidean case that we just saw; there are, however, other examples (for instance, the vertices might be bit strings and the distances their Hamming distance). Assuming that the distance function can be computed in constant time, the best known uniprocessor algorithm for this problem runs in (optimal) time proportional to N^2 on a graph of N vertices. The purpose of this paper is to present an algorithm for solving this problem on an N/(lg N)-processor parallel machine in time proportional to N lg N. (A similar result was achieved by Sollin [1977]; we will compare the new algorithm to his later in the paper.)

This paper is organized into three primary sections. In Section 2 we study a particular tree-structured parallel computer, and in Section 3 we will review a well-known algorithm for constructing MSTs. These sections provide the background for Section 4, in which we will see how the algorithm can be implemented very efficiently on the tree machine. Conclusions are then offered in Section 5.

2. Background on the Tree Machine

In this section we will examine the tree-structured searching machine described by Bentley and Kung [1979]. Although this discussion will provide the necessary background for applying the tree machine to the MST problem, it will be rather brief; the interested reader can refer to Bentley and Kung [1979] for additional details.

The basic structure of the tree machine is illustrated in Figure 1. It is a "mirrored" binary tree, containing three types of nodes: circles (which broadcast data), squares (which store data and compute), and triangles (which combine their inputs). To illustrate the operation of the tree machine, let us suppose that each of N square processors holds an element of set S, and we wish to determine whether 59 is a member of S. We enter 59 into the input node of the tree, and lg N steps later it reaches all the square nodes. We then perform N comparisons (in parallel), giving N bits telling whether the element stored in each processor is 59. We can now combine those bits in lg N steps by letting the triangles compute the logical or of their two inputs. We therefore receive the answer to our query 2 lg N steps after it was posed.

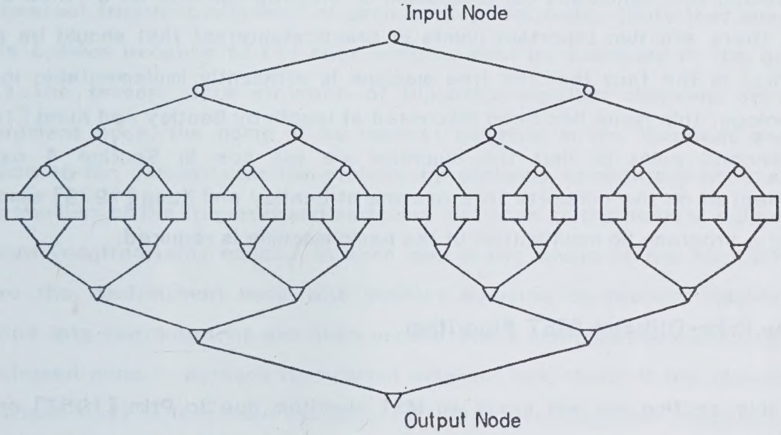


Figure 1. Structure of the tree machine.

The above example illustrates the important aspects of the tree machine. The primary function of the circles is to broadcast data (they also decide where a new record is to be inserted into the tree). The triangles' purpose is to combine their inputs; the operations they perform include *min*, *max*, *and*, *or*, and *plus*. In addition to returning the minimum value (for instance), simple bookkeeping enables them to return also the name of the object achieving it. The squares are the most complicated nodes. Each is a "baby" von Neumann computer with a small amount of main memory. They receive their instructions and data in a single stream transmitted by the circles. (The stream for the above member search would be something like "the following is data, constant 59, load into register A, load register B from the memory word holding the key, and compare.") An important property of the tree machine is that M successive member searches can be "pipelined" (that is, the second goes down the tree immediately following the first, and so on) to run in M-1 + 2 lg N time.

Note that the tree machine can be simulated to within a constant time factor by a set of N processors operating on a large shared memory with constant access time. Such a machine is a more typical theoretical model of parallel computation (see, for example, Borodin and Munro [1975] or Kung [1979]), and the result of Section 4 holds in this model as well as for the tree machine.

Although this concludes our discussion of the tree machine on a "theoretical" level, there are two important points of practical interest that should be made. The first is the fact that the tree machine is efficiently implementable in VLSI technology; this issue has been discussed at length by Bentley and Kung [1979]. The second point is that the algorithm we will see in Section 4 can be implemented on the concrete tree machine of Bentley and Kung [1979] simply by coding a program; no modification of the basic machine is required.

3. The Prim-Dijkstra MST Algorithm

In this section we will study an MST algorithm due to Prim [1957] and its computer implementation as given by Dijkstra [1959]. Because the algorithm has been examined at length in those pioneering papers (as well as in numerous texts), we will review it at a fairly intuitive level. Proofs of correctness and details of implementation abound in the literature.

Prim's algorithm works by starting with a *fragment* (subtree) containing one vertex of the graph and then successively expanding the fragment until a spanning tree is obtained. Because the fragment is expanded at each point under a very local condition, this is referred to as a "greedy" algorithm. To specify that condition, we must give some definitions. The *distance* of a vertex to the fragment is defined as the weight of the shortest edge connecting that vertex to the fragment. The *nearest neighbor of the fragment* is then defined as that vertex not in the fragment with minimum distance to the fragment (with ties broken arbitrarily). It is now easy to describe Prim's algorithm. The fragment is initialized to contain an arbitrary vertex. We then expand the fragment at each stage by adding to it the nearest neighbor of the fragment. When applying this algorithm to an N-node graph, after N-1 stages we will have emptied the set of nonfragment nodes and thereby have built a spanning tree of the graph. Prim gives an elegant proof that the resulting spanning tree is indeed minimum (the proof of this particular algorithm is based only on Principle 2 of his paper).

If Prim's algorithm is implemented in a computer program by keeping track of the "nearest nonfragment neighbor" of every fragment node, then the resulting algorithm has $O(N^3)$ running time. Dijkstra [1959] makes the brilliant observation that an $O(N^2)$ algorithm can be achieved by the dual strategy of keeping track of the "nearest fragment neighbor" of each nonfragment node. (Note that quadratic time is optimal because all $\binom{N}{2}$ edge weights must be examined in the general case.) The primary data structure of Dijkstra's algorithm contains, for each nonfragment node, the name of its nearest neighbor in the fragment and the distance to it. We initialize the program by choosing an arbitrary node as the single member of the fragment and assigning that node as the nearest neighbor of all other (nonfragment) nodes. In each successive stage of the algorithm, we choose the nonfragment node with minimum distance to nearest neighbor for insertion into the fragment and then update every nonfragment node to see if that chosen node is perhaps its nearest neighbor (we check if the distance to the chosen node is less than the distance to its current nearest neighbor). The informal method we have just seen is described more precisely as Algorithm 1.

Algorithm 1 assumes that the input vertices are named 1 through N and that the function Distance(i,j) returns the distance from node i to node j. The set TreeEdges contains pairs of integers, and its final value gives the MST. The set NonFragment contains the nonfragment nodes, each of which is represented as a Pascal record with three fields: Node (an integer node number), NNinFrag (an integer name of the node's nearest neighbor in the fragment), and NNDist (the distance to the node's nearest neighbor). Whitney [1972] gives a Fortran implementation of Algorithm 1 that implements the NonFragment set as an array; the running time of his program is proportional to N^2.

```
TreeEdges ← $\phi$
NonFragment ← $\phi$
for i ← 2 to N do
    New(a)
    a.Node ← i
    a.NNinFrag ← 1
    a.NNDist ← Distance(1,i)
    Insert a into NonFragment
while NonFragment $\neq \phi$ do
    Assign b s.t. b.NNDist = $\min_{c \in NonFragment}$ c.NNDist
    Insert (b.Node, B.NNinFrag) into TreeEdges
    Delete b from NonFragment
    foreach a in NonFragment do
        ThisDist ← Distance(a.Node, b.Node)
        if ThisDist < a.NNDist then
            a.NNinFrag ← b.Node
            a.NNDist ← ThisDist
```

Algorithm 1. The Prim-Dijkstra MST algorithm.

4. The Algorithm on the Tree Machine

In this section we will see how Algorithm 1 can be efficiently implemented on the tree machine that we studied in Section 2. We will use the tree machine to implement the NonFragment set; specifically, every record in the NonFragment set resides in a different square processor of the tree machine. We will assume that there is a small conventional computer "driving" the tree machine; that computer will also store the TreeEdges set. We will now describe how each part of Algorithm 1 can be implemented in this scheme.

Setting TreeEdges to be empty is trivial, and initializing NonFragment to be empty can be accomplished by broadcasting a "clear" command to all of the square processors in the tree machine. The for statement is implemented by performing the calculations in the "driver" computer, and then inserting each record into the tree machine. Since insertions can be pipelined, the total time required for this step is linear in N.

We now come to implementing the code in the while loop. We can find the record of NonFragment with the minimum NNDist field in logarithmic time by letting the triangles compute the minimum of those fields (and also returning the name of the record achieving that minimum). We then fetch the Node and NNinFrag fields of that record, insert the edge into TreeEdges, and delete the record from NonFragment (this amounts to turning off a processor). We must now perform the foreach statement, and it is here that we employ the massive parallelism of the tree machine. To every record currently stored in the tree, we (simultaneously) ship down a description of b.Node (sufficient to compute the distance function) and then compute ThisDist. If ThisDist is greater than the NNDist field of the particular processor, we (temporarily) turn it off. We now broadcast the assign commands to alter the NNinFrag and NNDist fields, which are executed only by the listening processors. We then turn back on all the processors currently representing elements on NonFragment, and repeat the while loop until NonFragment is empty.

The method that we just described is easy to program on the VLSI tree machine of Bentley and Kung [1979], assuming that the distance function between pairs of nodes can be computed by the square processors in constant time. The initialization steps (assigning sets to be empty and initializing NonFragment) require linear time on the tree machine. Each iteration of the while loop requires O(lg N) time. (Data must travel from the input node to the output node three times: once to find the minimum NNDist, once to find the associated values and delete the old record, and once to perform the updates of the foreach statement.) Since this step is executed N-1 times, the total time used by the step is O(N lg N). That cost dominates the cost of the initialization, and the total running time of the Prim-Dijkstra MST algorithm implemented on the tree machine is therefore proportional to N lg N.

To illustrate the operation of this parallel algorithm on a more concrete example, let us consider the problem of constructing the MST of a set of N points in k-space (using the Euclidean metric). Initializing the set NonFragment involves storing k reals in each of N processors; this can be accomplished in Nk time. In the while statement, finding the minimum can be accomplished in lg N time, and the name of the record realizing the minimum can also be retrieved (and deleted) that quickly. We must now execute the foreach statement. This involves entering the k coordinates of the point most recently added to the fragment to all of the N processors (and computing the sum of the squares of the coordinate-wise distances); this can be accomplished in time proportional to k. Updating the appropriate fields requires constant time, so the total time per step of this algorithm is proportional to k + lg N. The total running time of the algorithm is therefore proportional to N lg N + Nk.[1]

The MST algorithm of this section can be used to yield an N lg N time algorithm on a tree machine with N/(lg N) square processors, each of which has lg N storage. To accomplish this, we store lg N vertices at each square node, and implement the foreach statement by executing its code lg N times. Each iteration of the while loop still costs lg N time, and the total cost of the algorithm is still proportional to N lg N. The processor-time product of this modified algorithm is N^2, rather than the N^2 lg N of the previous version. Note that this scheme can be used to give an N·G(N) time MST algorithm on an N/G(N)-processor machine for any function G that grows more quickly than lg N (the trick is to store G(N) vertices in every square processor).

5. Conclusions

Sollin's [1977] algorithm is another method that has been proposed for the parallel construction of MSTs in N lg N time on an N-processor machine. The method that we saw in Section 4 is a fundamentally different approach to the problem than Sollin's: our method performs N iterations of logarithmic cost, whereas Sollin's performs lg N iterations of linear cost. A particular advantage of our method over Sollin's is that while his performs N^2 lg N distance calculations in the worst case, ours performs only $\binom{N}{2}$, which is optimal. A related advantage of ours is that it performs only N^2 operations altogether (unlike Sollin's N^2 lg N), and this approach might therefore lead to an N-processor algorithm with linear running time. We also saw that for any function G growing as fast as lg N, our algorithm can be implemented in N·G(N) time on an N/G(N)-processor machine, giving an optimal processor-time product of $O(N^2)$; this is superior to Sollin's processor-time product of $O(N^2 \lg N)$. Yet another parallel MST algorithm has been given by Savage [1977]. Her algorithm runs in $O(\lg^2 N)$ time on a machine with $O(N^2/\lg N)$ processors, and is therefore applicable in an entirely different sphere than the algorithm of Section 4. (Note that the processor-time product of her algorithm is proportional to $N^2 \lg N$.)

---

[1] It is possible to use the tree machine to achieve an asymptotically faster MST algorithm for point sets in k-space by implementing Yao's [1977] "directional nearest neighbor" MST algorithm. The directional nearest neighbors are found by pipelining the tree machine, and the resulting MST is then computed on a uniprocessor. The running time of the resulting algorithm is O[(N lg lg N + Nk)·f(k)], where f(k) is the number of directional nearest neighbor searches Yao's algorithm requires in k dimensions. The parallel implementation of the Prim-Dijkstra algorithm would probably be faster than this method in almost every practical application, however.

The contributions of this paper are in two spheres. In a "theoretical" setting, we have investigated an implementation of the Prim-Dijkstra MST algorithm on a parallel computer. While this algorithm has the same running time as a previous algorithm, it uses an optimal number of distance calculations, and therefore gives hope for a faster parallel algorithm. A modified version of the algorithm runs on N/G(N) processors, with optimal processor-time product. We also saw a faster parallel MST algorithm for the special case of finding the MST of a point set in Euclidean k-space, answering an open problem in computational geometry (see Shamos [1978, p. 210, Problem 7]). On a more practical level, the algorithm that we have investigated uses a machine with a simple and regular interconnection strategy that can be efficiently implemented in VLSI technology.

Acknowledgements

The helpful comments of Kevin Brown, Dorothea Haken, Jim Saxe, and Michael Shamos are gratefully acknowledged.

References

Bentley, J. L., B. W. Weide and A. C. Yao [1978]. "Optimal expected-time algorithms for closest-point problems," Carnegie-Mellon Computer Science Department Report CMU-CS-79-111 (March 1979).

Bentley, J. L. and H. T. Kung [1979]. "A tree machine for searching problems," to appear in the *1979 International Conference on Parallel Processing*.

Borodin, A. and I. Munro [1975]. *The Computational Complexity of Algebraic and Numeric Problems*, American Elsevier, New York.

Cheriton, D. and R. E. Tarjan [1976]. "Finding minimum spanning trees," *SIAM J. Computing* 5, 4, (December 1976), pp. 724-742.

Dijkstra, E. W. [1959]. "A note on two problems in connexion with graphs," *Numerisch Math. 1*, (October 1959), pp. 269-271.

Kung, H. T. [1979]. "The structure of parallel algorithms," to appear in *Advances in Computers*.

Prim, R. C. [1957]. "Shortest connection networks and some generalizations," *Bell System Technical Journal 36*, (November 1957), pp. 1389-1401.

Savage, C. D. [1977]. Parallel algorithms for graph theoretic problems. University of Illinois Coordinated Science Laboratory Report UILU-ENG-77-2231.

Shamos, M. I. [1978]. Computational Geometry, unpublished Ph.D. Thesis, Yale University, May 1978.

Sollin [1977]. An algorithm attributed to Sollin in *Introduction to the Design and Analysis of Algorithms* by S. E. Goodman and S. T. Hedetniemi (McGraw-Hill, 1977), Section 5.5. (Also referred to in Yao [1977].)

Whitney, V. K. M. [1972]. "Algorithm 422, minimal spanning tree [H]," *Comm. of the ACM 15*, 4 (April 1972), pp. 273-274.

Yao, A. C. [1977]. "On constructing minimum spanning trees in k-dimensional space and related problems," Stanford Computer Science Department Report STAN-CS-77-642 (December 1977).

PARALLEL ALGORITHMS AND SERIAL DATA STRUCTURES*
(Preliminary Draft)

J. IAN MUNRO
Dept. Computer Science, University of Waterloo
Waterloo, Ontario, N2L 3G1, Canada

EDWARD L. ROBERTSON
Dept. Computer Science, Indiana University
Bloomington, Indiana, 47401, USA

ABSTRACT

We consider data structures appropriate for use in an environment in which several processors have access to a single memory. In particular, it is shown that if k processors are available, an essentially k-fold speedup over the serial techniques can be obtained for the maintenance of a priority queue and of a pointer free structure of performing searches and updates.

1. INTRODUCTION

Our basic interest, in this paper, is in determining the extent to which one can take advantage of k parallel processors operating on a shared memory in maintaining basic data structures. It is well known, at least in the folk literature, that (log n)/log(k+1)# comparison steps are necessary and sufficient to find an element in an ordered list. (Here we assume comparisons are the basic measure of work, no arithmetic operations are permitted on the data elements, but on the other hand no change is made for resolving the interval in which an element lies once the necessary comparisons have been made.) This fact, together with the threat that fast shareable memories, if developed, are very likely to be considerably more expensive than singly accessible ones, sets a rather pessimistic tone. The observations that parallelism cannot be used to any great degree in following conventional pointers, and that protocols for modifying naturally accessable pointers tend to be costly, add to this pessimism.

On the other hand, there has been success in performing some basic tasks in a parallel environment. Valiant (6) has shown log log k time steps are necessary and sufficient to find the maximum of k elements using k comparators. The greatest success in the application of large numbers of processors to comparison based problems has been

---

*This work was supported in part by The Natural Science and Engineering Research Council of Canada under grant A8237.

#all logarithms are taken to base 2

in sorting ((1)(5)). The contribution of this paper is to continue this study of basic operations on implicit (i.e., essentially pointer free) structures. The results are all positive as we demonstrate that, by employing k processors, an essentially k-fold speedup can be achieved over the serial algorithms for the implementation of a priority queue and a structure for performing insertions, deletions and searches. We examine the priority queue, before turning to the more difficult general structure.

2. PRIORITY QUEUES

Under the serial model, about log n comparisons are sufficient to insert an element into an implicit priority queue (i.e. a heap) and about twice that number suffice to remove the maximum element and reconstitute the structure ((2)(7)). Indeed these costs are O(1) and log n + O(1) on the average, corresponding to the information theory lower bound of log n for the insert/extract pair of operations. We can obtain almost the full k-fold speedup in a parallel processing environment.

Theorem - Using k processors and O(k) pointers and extra storage locations, an implicit priority queue can be maintained at a maximum cost of (log n)/k + O(1) comparison steps per insertion and 2(log n)/k + O(1) for extracting the maximum. The average cost for insertion is O(1) and for extracting the maximum, (log n)/k + O(1). This corresponds to the average case lower bound of (log n)/k - O(1) for an insert/extract pair.

Outline of Proof - We will sketch the basic scheme for maintaining a priority queue which accepts a new request whenever possible. From this construction, the precise worst case behaviour suggested in the theorem is easily achieved.

The scheme is essentially to maintain a conventional heap (with maximum in location 1) and to coordinate the operation of the usual serial algorithm in a parallel processing environment. Insertions are made by incrementing n (the number of elements in the system) putting the new element in location n and allowing it to "sift up". Extracting the maximum is achieved by removing the largest element in the system, which is in location 1, and letting this empty location "sift down" to a leaf. The element in location n is moved to the vacant leaf, n is decremented, and the displaced element filters up. The basic scheme is, then, the usual serial processing algorithm for maintaining a heap ((2)(7)). The bounds in a serial processing environment are well known, our task is to outline the coordination scheme.

We will not complete a transaction before accepting the next; each insertion or maximum extraction is (at any given time) associated with its own processor. The next request can be accepted after the preceding one has had a few steps "head start" and a processor is available.

Sifting up: There are two difficulties in having several elements sifting up at the same time. The first is that of two "brothers" trying to displace their father at the same time. The second is having a father and son in motion simultaneously. In such a case it seems possible that we may try to simultaneously promote and demote an element. We also observe that if two elements begin sifting up at consecutive time steps, they could be at the same depth in the tree as they move up. By insisting that the sifting up process be initiated no sooner than three time steps after the preceding one, we guarantee no such coordination difficulties occur.

Finding the maximum: We will insist that the largest element in the structure is always in location 1, even though elements may be sifting up. Each processor must retain a pointer to the element which it is sifting up or the empty slot it is moving down. The convention is maintained that the maximum element is in location 1, and that processor i (and so pointer i) is concerned with the i^{th} largest element sifting up (or waiting to be sifted up as outlined in the preceding section). Therefore, when an element is to be sifted up (i.e. either as a new insertion or as a result of being moved from the end of the structure after a maximum extraction), it is compared with all the elements of this status, and the element in location 1. The necessary adjustments to the pointers are made before the next comparison step. This operation may require one time step of all the processors. It is not counted as a step for synchronization purposes and no other operations occur at the same time, even if processors are available.

When the maximum element is extracted, the larger of the elements in locations 2 and 3 is promoted (other processing continues for this step). This new element in location 1 is then compared with all other elements in (or waiting to enter) the sift up phase (all other processing waits a step and this is not counted for synchronization purposes). The appropriate adjustments are then made before further processing resumes.

Sifting down: The only problem in sifting an empty location down in the structure is that of a location and its father moving down at the same time. Insisting that a maximum extraction be initiated at least two steps after the preceding one, solves this problem.

En passant: There remains the minor issue of an empty location sifting down encountering an element sifting up. If an element moving up is compared with a "blank", it awaits the (simultaneous) outcome of the comparison of the element and its brother. If this results in the promotion of the element it continues sifting up, otherwise it stops.

3. A STRUCTURE FOR FAST RETRIEVAL AND UPDATE

Another interesting structure, at least in our view, is the implicit structure for performing insert, delete and search. Munro and Suwanda (3) show that if no pointers are maintained and the data elements in n locations are required to be in a fixed partial order, then $O(\sqrt{n})$ basic operations of compare and move are necessary and sufficient to maintain the structure. Again we can obtain essentially a full k-fold speed-up.

Theorem - $O(\sqrt{n}/k)$ basic comparison and move steps are necessary and sufficient to maintain an implicit structure in a fixed partial order for performing the operations insert, delete and search.

Essence of Proof - The structure maintained is that suggested in (3), the algorithms, however, are somewhat different. Elements are stored in what can be viewed as the upper left half of a square array, which is sorted by row and by column (the maximum in the top left).

In (3) a $3\sqrt{2n}$ comparison search algorithm is suggested for the serial processing environment, this is improved to the lower bound for the structure, $2\sqrt{2n}$, in (4). The former scheme, however, is the one more directly amenable to parallel processing, as it is a divide and conquer scheme. The basic operation is that of performing a binary search on a row of column of a substructure and thus being able to reject essentially half of the elements in the old structure as being too large or too small. At this point we are left with two smaller structures to search, clearly this is the ideal situation for parallel processing. Even with a rather restricted view of the type of processor coordination which can go on between comparisons steps, the search can easily be performed in $3\sqrt{2n}/k + o(\sqrt{n}/k)$ comparison steps.

The modification schemes in a parallel processing environment differ more substantially from the serial proceeding one. Viewing the left and top neighbours of an element as its parents we note that an element has at most i+1 ancestors at the i^{th} preceding generation (two parents, three grandparents etc.). When a new element is to be inserted it is placed in location n+1. We then determine its minimum ancestor $(\sqrt{2n}/k)$ generations back. If this element is smaller than the new element, it is swapped with it and the process is continued on the new element from its new position. By the time this process terminates, the structure may be in a state of disarray (pardon the pun). Using one processor for each displaced element, including the new one, these elements are permitted to sift up (as in a conventional heap). The key is that they cannot move up more than $\sqrt{2n}/k$ levels. Deletions are handled in an analogous manner.

It is shown in (3) that the product of the search time and update time in the serial processing environment is at least n provided the elements are kept in some fixed partial order. The theorem follows from this fact and the scheme outlined above.

4. CONCLUSION

We have demonstrated that two data structures, originally designed for serial processing, are amenable to the parallel environment. Indeed with the use of k processors we achieve essentially a k-fold speed-up. It would appear that further study of implicit data structures in a parallel processing environment will shed more light on the power and limitations of parallelism.

5. REFERENCES

(1) Batcher, K.E., "Sorting Networks and their Applications", Proc. AFIPS Spring Joint Computer Conference., Vol. 32, April 1968, pp. 307-314.

(2) Floyd, R.W., "Algorithm 245: Treesort 3, CACM 7,12 (Dec 1964), p. 701.

(3) Munro, J.I. and H. Suwanda, "Implicit Data Structures", Proc. 11th Annual ACM Symp. on Theory of Computing, May 1979, pp. 108-117.

(4) Munro, J.I., and H. Suwanda, "Implicit Data Structures for Fast Retrieval and Update", University of Waterloo, Computer Science Dept., Research Report CS-79-31.

(5) Preparata, F.P., "New Parallel-Sorting Schemes", IEEE Trans. on Computers, C-27,7 (July 1978), pp. 669-673.

(6) Valiant, L.G., "Parallelism in Comparison Problems", SIAM J. on Comp. 4,3 (Sept 1975), pp. 348-355.

(7) Williams, J.W.J., "Algorithm 232: Heapsort", CACM 7,6 (June 1964), pp. 347-348.

PARALLEL MATRIX AND GRAPH ALGORITHMS*

ELIEZER DEKEL, DAVID NASSIMI, AND SARTAJ SAHNI
University of Minnesota
Minneapolis, Minnesota 55455

Abstract: Matrix multiplication algorithms for cube connected and perfect shuffle computers are presented. It is shown that in both these models, two n x n matrices can be multiplied in $0(\frac{n}{m} + \log m)$ time when $n^2 m$, $1 \le m \le n$ processing elements (PEs) are available. When only m^2, $1 \le m \le n$, PEs are available, two n x n matrices can be multiplied in $0(n^2/m + m\,(n/m)^{2.795})$ time. It is shown that many graph problems can be solved efficiently using the matrix multiplication algorithms.

Keywords and phrases: cube connected computer, perfect shuffle computer, parallel algorithm, matrix multiplication, graph algorithms, complexity.

1. Introduction

Computer systems may be broadly classified into four categories (Flynn [6]). These are; i) SISD (Single Introduction Stream, Single Data Stream); ii) SIMD (Single Instruction Stream, Multiple Data Stream); iii) MISD (Multiple Instruction Stream, Single Data Stream) and iv) MIMD (Multiple Instruction Stream, Multiple Data Stream). This paper is concerned solely with SIMD type computers. A SIMD type computer system is comprised of some number, p, of processing elements (PEs), each having some local memory. We shall assume that the PEs are indexed 0 through p-1 and shall refer to the ith PE as PE(i). The PEs are synchronized and operate under the control of a single instruction stream. An enable/disable mask may be used to select a subset of PEs that will perform the instruction to be executed at any given time. All enabled PEs perform the same instruction. Two SIMD architectures will be relevant to our discussion:

i) Cube Connected Computers (CCC)

Assume that $p=2^q$ and let $i_{q-1} \ldots i_0$ be the binary representation of i for i $\epsilon[0,p-1]$. Let $i^{(b)}$ be the number whose binary representation is $i_{q-1} \ldots i_{b+1} \bar{i}_b i_{b-1} \ldots i_0$ where $\bar{i}_b$ is the complement of i_b and $0 \le b < q$. In the cube model, PE(i) is connected to PE($i^{(b)}$), $0 \le b < q$. Data can be transmitted from one PE to another only via the interconnection pattern.

ii) Perfect Shuffle Computer (PSC)

Let p, q, i and $i^{(b)}$ be as in the cube model. In the perfect shuffle model, PE(i) is connected to PE($i^{(0)}$), PE($i_{q-2} i_{q-3} \ldots i_0 i_{q-1}$) and PE($i_0 i_{q-1} i_{q-2} \ldots i_1$). These three connections will respectively be called exchange, shuffle and unshuffle. Once again, data transmission from PE to PE is possible only via the connection scheme.

The reader is referred to [1], [2], [3], and [7] for related work on matrix algorithms.

---

*This research was supported in part by NSF grant MCS 78-15455. David Nassimi is currently with the department of Electrical Engineering and Computer Science, Northwestern University, Evanston, Illinois 60201.

2. Cube Connected Computer

In presenting our algorithms, we shall make use of the following conventions and notations:

(i) "←" will be used to denote an assignment requiring data movement between PEs that are directly connected. ":=" will be used to denote an assignment in which all variables in the left and right hand side of ":=" are local to the same PE or to the control unit.

(ii) We shall use i_r to denote bit r in the binary representation of i. Thus the binary representation of i is $i_{u-1}i_{u-2}\ldots i_0$ for some u. $i^{(b)}$ will denote the number with binary representation $i_{u-1}i_{u-2}\ldots i_{b+1}\bar{i}_b i_{b-1}\ldots i_0$ where $\bar{i}_b=1-i_b$.

(iii) PE(k) will denote the PE with index k. Similarly, A(k), B(k), C(k), etc. will denote memory locations or registers in PE(k).

(iv) PEs may be enabled by providing a selectivity function. This function can be placed after any statement. Operations are performed only on enabled PEs. For example, the statement

$$A(i) := B(i) + C(i), (i_2=0)$$

has the selectivity function $(i_2=0)$. As a result, the addition and assignment is carried out only on PEs whose index i has bit 2 equal to 0. The statement

$$A(i^{(b)}) \leftarrow B(i),\ (i_0=1)$$

specifies a data route. Data from the B register of PEs with index i and $i_0=0$ is routed to the A register of corresponding PEs with bit b equal to $1-i_b$.

(v) The complexity of an algorithm is the sum of the PE time needed and the communication time (i.e. the time needed to route data from PE to PE). A unit-route is a data transmission from a PE to another PE to which it is directly connected.

2.1 CCCs with n^3 PEs

Consider a CCC with $n^3 = 2^{3q}$ PEs. Conceptually, these PEs may be regarded as arranged in an n x n x n array pattern. Assuming that the PEs are indexed in row major order, the PE, PE(i,j,k) in position (i,j,k) of this array has index $in^2 + jn + k$ (note that array indices are in the range [0,n-1]). Hence, if $r_{3q-1},\ldots,r_0$ is the binary representation of the PE position (i,j,k) then $i=r_{3q-1}\ldots r_{2q}$, $j=r_{2q-1}\ldots r_q$ and $k=r_{q-1}\ldots r_0$. Using A(i,j,k), B(i,j,k) and C(i,j,k) to represent memory locations in PE(i,j,k), we can describe the initial condition for matrix multiplication as:

$$\left.\begin{array}{l} A(0,j,k) = a_{jk} \\ B(0,j,k) = b_{jk} \end{array}\right. \quad 0 \le j < n \text{ and } 0 \le k < n \quad .$$

a_{jk} and b_{jk} are the elements of the two matrices to be multiplied. The desired final configuration is:

$$C(0,j,k) = c_{jk} \quad 0 \le j < n \text{ and } 0 \le k < n$$

where

$$c_{jk} = \sum_{\ell=0}^{n-1} a_{j\ell} b_{\ell k} \tag{2.1}$$

Our algorithm computes the product matrix C by directly making use of (2.1). The algorithm has three distinct phases. In the first, elements of A and B are distributed over the n^3 PEs so that we have $A(\ell,j,k)=a_{j\ell}$ and and $B(\ell,j,k)=b_{\ell k}$. In the second phase the products $C(\ell,j,k)=A(\ell,j,k)* B(\ell,j,k) = a_{j\ell}\, b_{\ell k}$ are computed. Finally, in the third phase the sums $\sum_{\ell=0}^{n-1} C(\ell,j,k)$ are computed. The details are spelled out in procedure CCM1 (Algorithm 2.1). In this procedure all PE references are by PE index (recall that the index of PE(i,j,k) is in^2+jn+k). Lines 1-10 implement phase 1. The loop of lines 1 to 4 copies the data initially in PE(0,j,k) to the PEs, PE(i,j,k), $1 \le i < n$ (recall that bits 3q-1,...,2q of a PE index yield the i value). Following this loop, we have:

$$\left.\begin{array}{l} A(i,j,k)=a_{jk} \\ B(i,j,k)=b_{jk} \end{array}\right\} ,\ 0 \le i < n.$$

Note that $A(i,j,i)=a_{ji}$. In lines 5 to 7, A(i,j,k) is replicated over A(i,j,k), $0 \le k < n$ with the result that $A(i,j,k) = a_{ji}$, $0 \le k < n$. The loop of lines 8-10 replicates $B(i,i,k)=b_{ik}$ over B(i,j,k), $0 \le j < n$. In line 11, the product $C(i,j,k) = A(i,j,k)*B(i,j,k)=a_{ji}b_{ik}$ is computed in PE(i,j,k), $0 \le i < n$, $0 \le j < n$ and $0 \le k < n$. Finally, the loop of lines 12-14 computes the sum

$$C(0,j,k) = \sum_{i=0}^{n-1} C(i,j,k) = \sum_{i=0}^{n-1} a_{ji}b_{ik}=c_{jk}.$$

The analysis of CCM1 is quite straightforward. The processor communication time in each of the last three for loops is that for q unit-routes. The first for loop requires 2q unit-routes. Hence, the total communication time is that for 5q=O(logn) unit-routes. The PE time is clearly seen to be O(logn).

```
line  procedure CCM1(A,B,C)
        //CCC multiplication algorithm·$n^3$ PEs//
  1     for $\ell$:=3q-1 down to 2q do
  2         $A(j^{(\ell)}) \leftarrow A(j)$, $(j_\ell=0)$
  3         $B(j^{(\ell)}) \leftarrow B(j)$, $(j_\ell=0)$
  4     end
  5     for $\ell$:=q-1 down to 0 do//replicate A(i,j,i) into A(i,j,*)//
  6         $A(j^{(\ell)}) \leftarrow A(j)$, $(j_\ell = j_{2q+\ell})$
  7     end
  8     for $\ell$:=2q-1 down to q do//replicate B(i,i,k)//
  9         $B(j^{(\ell)}) \leftarrow B(j)$, $(j_\ell=j_{q+\ell})$
 10     end
 11     C(j):= A(j) * B(j)
 12     for $\ell$:=2q to 3q-1 do //add terms//
 13         $C(j) \leftarrow C(j) + C(j^{(\ell)})$
 14     end
 15   end CCM1
```

Algorithm 2.1

2.2 CCCs with n^2 PEs

Two $n \times n$ matrices can be multiplied in $O(n)$ time on a CCC with n^2 PEs. Our CCC multiplication algorithm has the same two phases as does Cannon's [2]. However the initial alignment and later shifting pattern is substantially different. We assume that $n^2=2^{2q}$ PEs are available. Conceptually, these PEs may be viewed as in an $n \times n$ array (see Figure 2.1). Once again, we shall use two different notations to refer to the jth PE. PE index is hardware determined. A(j), B(j), C(j) will refer to memory locations or registers in PE(j). In the $n \times n$ array, we assume that the index of the PE in position (i,j) is the same as the row-major index of the position. Thus if PE(k) is at position (i,j) of the array then $k=in+j$. So, we can, without confusion, refer to the PE in position (i,j) as PE(i,j). Similarly, we can use the notation A(i,j), B(i,j) and C(i,j). Our algorithm however will only use the notation A(j), B(j), C(j).

i \ j	0	1	2	3
0	0	1	2	3
1	4	5	6	7
2	8	9	10	11
3	12	13	14	15

Figure 2.1 Array view of a 16 PE CCC. Each square represents a PE. The number in a square is the PE index.

Procedure CCM2 (Algorithm 2.2) is a formal statement of our multiplication algorithm. It is assumed that A and B have been initialized so that $A(i,j)=a_{ij}$ and $B(i,j)=b_{ij}$, $0 \le i < n$, $0 \le j < n$. The algorithm computes $C(i,j)=c_{ij} = \sum_{k=0}^{n-1} a_{ik} b_{kj}$, $0 \le i < n$, $0 \le j < n$. Lines 1-5 obtain the initial alignment. Lines 6-13 form the product matrix by repeated shift-multiply-add steps. The loop of lines 1-5 results in the following configuration (Lemma 2.1):

$$\begin{aligned} A(i,t) &= a_{i,i \oplus t} \\ B(i,t) &= b_{i \oplus t,\, t} \end{aligned} \tag{2.2}$$

$i \oplus t$ denotes an exclusive or of the binary representations of i and t. Hence, the initial product of line 6 computes $C(i,t) = a_{i,i \oplus t} * b_{i \oplus t,t}$ which is one of the terms in the sum for c_{it} (see eq.(2.1)). FUNC (line 8) is a function that controls the shifting of A and B values. It returns a bit index in the range $[0,q-1]$ (recall that $n=2^q$). Note that the q least significant bits in a PE index (using the one dimensional indexing scheme) determines the column index for the two dimensional indexing scheme while the q most significant bits determine the row index. Keeping this in mind, we easily see that line 10 shifts the elements in A along rows while line 11 shifts elements of B along columns. Thus in line 12 we always have $A(i,t)=a_{i,p}$ and $B(i,t)=b_{r,t}$ for some p and r. Lemma 2.2 shows that r=p always so that we get a valid pairing in A(i,t) and B(i,t). From the construction of FUNC (yet to be specified) it will follow that no a or b value can show up in the same PE more than once. Hence, each time the product A(j)*B(j) is computed in line 12, a new term

```
line  procedure CCM2(A,B,C)
        //multiply two n x n matrices on a CCC with $n^2$ PEs//
        //obtain initial alignment//
  1     for $\ell$:= 0 to q-1 do
  2         k:= q + $\ell$
  3         $A(j) \leftarrow A(j^{(\ell)}),\ (j_k=1)$
  4         $B(j) \leftarrow B(j^{(k)}),\ (j_\ell=1)$
  5     end
  6     C(j) := A(j) * B(j)
  7     for k:= 1 to n-1 do //shift-multiply//
  8         $\ell$:= FUNC(k) //get a bit index $\ell \in \{0,1,\ldots,q-1\}$//
  9         m:= q + $\ell$
 10         $A(j) \leftarrow A(j^{(\ell)})$
 11         $B(j) \leftarrow B(j^{(m)})$
 12         C(j) :=C(j) + A(j)*B(j)
 13     end
 14   end CCM2
```

Algorithm 2.2

of eq.(2.1) is formed. This implies that when the loop of lines 7-13 terminates, the product matrix will have been computed.

Lemma 2.1: When the loop of lines 1-5 of CCM2 terminates, eq.(2.2) correctly represents the values for $A(i,t)$ and $B(i,t)$, $0 \leq i < n$, $0 \leq t < n$.
Proof: See [5].

Lemma 2.2: Whenever line 12 of CCM2 is reached, $A(i,t)=a_{ir}$ and $B(i,t)=b_{rt}$ for some r (note that $j = in+t$), $0 \leq i < n$ and $0 \leq t < n$.
Proof: See [5].

We now proceed to specify FUNC. We are looking for a function that will generate a sequence of n-1 integers in the range [0,q-1] such that by successively complementing along each bit in this sequence, all elements of a row will pass through each PE in that row (and so by adding q to each element in the sequence, we can get every element in a column to go through each PE in that column). In other words, in terms of the notation of Lemma 2.2, we want the sequence of integers [5].

$$\{z^{\ell_1}, z^{\ell_2,\ell_1}, \ldots, z^{\ell_{n-1},\ldots,\ell_1}\}$$

to be some permutation of $\{1,2,\ldots,n-1\}$. It is very easy to arrive at a suitable FUNC if we think recursively: Divide a row into two halves. All PEs in the left half have bit q-1 = 0 while those in the right half have bit q - 1 = 1. If we have a bit sequence S_{q-2} that can be used when the number of PEs in a row is 2^{q-1} then this sequence will cause all elements in each half of a 2^q PE row to go through each PE in that half. By complementing on bit q-1 we can exchange halves and then reuse S_{q-2} to make each element go through the PEs it hasn't already gone through. So, we obtain:

$$S_{q-1} = S_{q-2}, q-1, S_{q-2}$$

and $S_0 = 0$

When q=4, the sequence to use is

$$\begin{aligned} S_3 &= S_2,3,S_2 \\ &= S_1,2,S_1,3,S_1,2,S_1 \\ &= S_0,1,S_0,2,S_0,1,S_0,3,S_0,1,S_0,2,S_0,1,S_0 \\ &= 0,1,0,2,0,1,0,3,0,1,0,2,0,1,0 \end{aligned}$$

It should be easy to see how FUNC may be incorporated into CCM2, in-line, using only $0(q)=0(\log n)$ control unit memory. The time needed to generate the entire sequence is $0(n)$.

As far as the complexity of CCM2 is concerned, exactly $2(q+n-1)$ unit-routes are needed. Both the PE and communication time are $0(n)$.

2.3 CCCs with n^2m, $1 \leq m \leq n$ PEs

By combining together algorithms CCM1 and CCM2 one can obtain an efficient parallel multiplication algorithm for CCCs with n^2m, $1 \leq m \leq n$, PEs. We shall assume that n and m are powers of 2. So, m divides n. The matrices, A and B, to be multiplied can each be partitioned into m^2 $n/m \times n/m$ equal sized submatrices A_{ij} and B_{ij}, $0 \leq i < m$ and $0 \leq j < m$. The product matrix C may be similarly partitioned. It is easy to see that C_{ij} is given by:

$$C_{ij} = \sum_{j=1}^{n} A_{ik\;\;kj}, \quad 0 \leq i < m,\ 0 \leq j < m \qquad (2.3)$$

The partitioned matrices may be viewed as an mxm matrix with each element being an $n/m \times n/m$ matrix. The n^2m PEs in the CCC may be viewed as forming an mxnxn array $PE(i,j,k)$, $0 \leq i < m$, $0 \leq j < n$ and $0 \leq k < n$. For each fixed i, the PEs in this array form an nxn array which can be partitioned into m^2 $n/m \times n/m$ arrays $PE_{i,j,k}$, $0 \leq j < m$, $0 \leq k < m$. The initial configuration for matrix multiplication is given by $A_{0,j,k}=A_{j,k}$ and $B_{0,j,k}=B_{jk}$, $0 \leq j < m$, $0 \leq k < m$. $A_{0,j,k}$ and $B_{0,j,k}$ denotes the A and B registers of the PE partition $PE_{0,j,k}$. A and B can be multiplied using (2.3), CCM1 and CCM2. First, the partitioned submatrices of A and B are replicated as in lines 1 to 10 of CCM1 to obtain $A_{i,j,k}=A_{j,i}$ and $B_{i,j,k}=b_{ik}$, $0 \leq i < m$, $0 \leq j < m$ and $0 \leq k < m$. This replication requires only $0(\log m)$ time. The product of line 13 of CCM1 is now a product of two $n/m \times n/m$ matrices A_{ji} and B_{ik}. The number of PEs available to carry out each product $A_{ji}*B_{ik}$ is n^2/m^2 (i.e. the PE partition $PE_{i,j,k}$). So, CCM2 can be used to implement line 11 of CCM1 in $0(n/m)$ time. Finally the addition loop of lines 12 to 14 (which perform the summation of (2.3)) is easily implemented in $0(\log m)$time.

2.4 CCCs with m^2 PEs, $1 \leq m \leq n$

The partitioning strategy of section 2.3 can also be used when only m^2, $1 \leq m \leq n$ PEs are available. In this case A and B are partitioned into $A_{i,j}$ and $B_{i,j}$, $0 \leq i < m$ and $0 \leq j < m$ as before. The PEs are viewed as forming an mxm array $PE(i,j)$, $0 \leq i < m$ and $0 \leq j < m$. Initially, $PE(i,j)$ contains the n^2/m^2 elements of $A_{i,j}$ as well as the n^2/m^2 elements of $B_{i,j}$. The matrix product can be formed proceeding essentially as in procedure

CCM2 with n and q respectively replaced by n/m and log n/m. The routes of lines 3, 4, 10 and 11 of CCM2 now involve n^2/m^2 elements each and the product of line 6 is a product of two n/m x n/m matrices. This product takes $O(n^3/m^3)$ time using the classical matrix multiplication algorithm. If Pan's [10] matrix multiplication algorithm is used then line 6 takes $O((n/m)^{\beta})$ time where $\beta = 1 + \log 2052/\log 70 \approx 2.795$. Hence the overall complexity of CCM2 when adapted to the case of m^2 PEs is $O(\frac{n^2}{m} + m(\frac{n}{m})^{\beta}) = O(\frac{n^2}{m} + \frac{n^{2.795}}{m^{1.795}})$.

The interesting special case of n PEs falls into the case just discussed when $m=\sqrt{n}$ (provided of course n is a perfect square). Actually, for the case of n PEs there exists a very straightforward $O(n^2)$ algorithm for matrix multiplication. The algorithm uses the classical matrix multiplication algorithm and starts with column i of A and B in PE(i). We leave it to the reader to fill in the details of this algorithm.

3. Perfect Shuffle Computer

The CCC algorithms can be adapted to run on a PSC in the same asymptotic time as on a CCC. The details of the adaptation can be found in [5].

4. Applications to Graph Problems

Efficient parallel algorithms for several graph problems may be obtained using the matrix multiplication algorithms of sections 2 and 3. We discuss some of these in the following sub-sections. Analyses are provided only for the case of n^3 PEs. The following discussion assumes we are dealing with n vertex graphs. Central to many of these applications is the repeated squares method of computing the transitive closure of a boolean matrix (i.e. $A^*=(A+I)^n=((A+I)^2)^2....)^2)$. Since this requires $\log n$ matrix multiplications, transitive closures may be determined in $O(\log^2 n)$ time on n^3 PE PSCs and CCCs.

4.1 All pairs shortest paths

The all pairs shortest path matrix A, is an n x n matrix such that A(i,j) is the length of a shortest path from i to j. Let $A^k(i,j)$ denote the length of a shortest path from i to j going through at most k intermediate vertices. Clearly, $A(i,j)=A^n(i,j)$. Let $A^0(i,j)$ be the length of the edge <i,j> if <i,j> is in the graph. Let $A^0(i,j)$ be $+\infty$ if < i j> is not in the graph and 0 if i=j. It should be easy to see that:

$$A^k(i,j)=\min \{A^{k/2}(i,\ell) + A^{k/2}(\ell,j)\}$$

Hence, A^n may be computed from A^0 by computing $A^2, A^4, A^8, \ldots, A^n$. A^k may be computed from $A^{k/2}$ using the matrix multiplication algorithm with + substituted for * and min substituted for +. The complexity of the resulting algorithm is $O(\log^2 n)$ on an n^3 PE PSC or CCC.

4.2 Radius, Diameter and Centers

The radius, diameter and centers of a graph may be trivially computed in

$O(\log n)$ time (on an n^3 PE PSC or CCC) from the all pairs shortest path matrix (which itself requires $O(\log^2 n)$ to compute).

4.3 Median and Median Length

Let $d(i,j)$ be the length of a shortest path from i to j. Let $h(i)$ be the weight of vertex i. Vertex v is a weighted median of the graph iff:

$$\sum_{j=1}^{n} h(j)d(v,j) \leq \sum_{j=1}^{n} h(j)d(k,j) \quad , \quad 1 \leq k \leq n.$$

When $h(j) = 1$, $1 \leq j \leq n$, vertex v is simply a median of the graph. $\sum_{j=1}^{n} h(j)d(v,j)$ is called the weighted median length of the graph. For any graph these quantities can be easily computed once the shortest path matrix A has been determined. The sum $\sum_{1}^{n} h(j)d(k,j) = \sum_{1}^{n} h(j)A(k,j)$ can be computed for all k in $O(\log n)$ time on an n^3 PE PSC and CCC. The minimum of these can also be found in this much time.

4.4 Shortest Path, Breadth First, Minimum Depth and Least Spanning Trees

A shortest path spanning tree with root v is a spanning tree of the given graph. Its root is vertex v and the distance from v to each vertex j equals $d(v,j)$. ($d(v,j)$ is the shortest distance from v to j in the graph.) Let $R(i,j)$ be a vertex index such that $R(i,j)=0$ or $A(i,R(i,j))+A(R(i,j),j)= A(i,j)$. $R(i,j)=0$ iff the shortest i to j path has no intermediate vertices. Otherwise $R(i,j)$ gives a vertex that is half way along the i to j path (i.e. if there are p intermediate vertices then $R(i,j)$ is either the $\lfloor p/2 \rfloor$th or $\lceil p/2 \rceil$th vertex). Clearly, $R(i,j)$ can be computed in $O(\log^2 n)$ time along with the computation of $A(i,j)$. A shortest path spanning tree rooted at v may be obtained from $R(i,j)$ in $O(\log^2 n)$ time. If $R(v,j)=0$ then j is a child of v and we may set PARENT(j)=v. If $R(i,j)=k$ then j is a descendent of k and we need to examine $R(k,j)$. $R(k,j)$ can be routed to PE(i,j) in $O(\log n)$ time. If $R(k,j)=0$ then set PARENT(j)=k. Otherwise j is a descendent of $R(k,j)$. Since $R(i,j)$ is midway from i to j and $R(k,j)$ midway from $R(i,j)$ to j etc., we will have to follow down at most $\log n$ R values before discovering the parent of j. Hence, all nodes will know their parents in the spanning tree in $O(\log^2 n)$ time plus the $O(\log^2 n)$ time needed to compute R initially.

A breath first spanning tree can be obtained in the same way as above by starting with $A^0(i,j)=1$ if (i,j) is an edge in the graph, $A^0(i,j)=\infty$ if (i,j) not in the graph and $A^0(i,j)=0$ if $i=j$. Note that $A(v,j)$ is the depth (less one) of node j in the breadth first spanning tree with root v. Hence, the depth $D(v)$ of the spanning tree rooted at vertex v is $\max\{A(v,j)\}+1$. A minimum depth spanning tree may be obtained in $O(\log^2 n)$ time by first computing A as for a breadth first spanning tree, then compute $D(j)$ for all j (this takes only $O(\log n)$ time). Next, $\min(D(j))$ is computed (again only $O(\log n)$ time is needed). The j value with minimum $D(j)$ is the root of the minimum depth spanning tree. Now find the breadth first spanning tree with root j. A shortest median spanning tree may be found similiarly.

4.5 Max Gain

Let G be a directed acyclic graph. Let $w(i,j) \geq 0$ be the weight of edge $<i,j>$ (assume $w(i,j)=0$ if $<i,j>$ not in G). The gain on an i,j path is the product of the edge weights on that path. W is the max gain matrix iff $W(i,j)$ is the maximum gain for every pair i,j. Clearly, this matrix may be computed in $O(\log^2 n)$ time using an approach similar to that used for the all pairs shortest path matrix.

4.6 Topological Sort and Critical Paths

These problems are described in [9]. Given an acyclic directed graph $G=(V,E)$, we are required to list the vertices in topological order. This can be done in $O(\log^2 n)$ time using matrix multiplication. Let $A^0(i,j)=1$ if $<i,j> \in E(G)$, $A^0(i,j)=-\infty$ if $<i,j> \notin E(G)$ and $A^0(i,i)=0$, $0 \leq i \leq n-1$. Use the all pairs shortest path algorithm to compute the lengths of the longest paths. This requires using max in place of the min used on the all pairs shortest path algorithm. Set $A(0,i)=0$ for all i with the property that $A(t,i) \leq 0$, $0 \leq t < n$. Note that now $A(0,i)=0$ for exactly those vertices in G that have no predecessors. Also note that $A(i,i)>0$ for some i iff G has a directed cycle. This is not possible as G is acyclic. Let $i_1, i_2, \ldots, i_k$ be the values of i for which $A(0,i)=0$. For each j for which $A(0,j) \neq 0$, compute $A(0,j) = \max_{1 \leq p \leq k} \{A(i_p,j)\}$. This can clearly be done in $O(\log n)$ time on both an n^3 PE PSC and CCC. $A(0,j)$ gives the length of the longest path from any node with no predecessors to node j. Now sort the pairs $[A(0,j),j]$, $0 \leq j < n$ into nondecreasing order of $A(0,j)$. This can be done in $O(\log n)$ time on an n^3 PSC or CCC using the algorithm of [11]. If the pair $[A(0,j),j]$ ends up in $PE(0,k)$ then vertex j is the kth vertex in the topological order. The correctness of this statement is easily verified.

The strategy just described can also be used to determine early and late start times for tasks in an activity on edge network (see [9]). In such a network each edge represents an activity and has a length associated with it. The early start time for any activity $<i,j>$ is the length of the longest path in the given directed acyclic graph from the start vertex, 0, to vertex i. The early start times can easily be found in $O(\log^2 n)$ time using the first part of the above strategy. Once the early start times are known, the earliest time the project can be finished is known. From this the latest start times (i.e. the time by which an activity must start so that the project length doesn't increase) can be computed in another $O(\log^2 n)$ steps. Knowing both the early and late start times the critical activities and critical paths are readily obtainable.

From the discussion of the preceding six subsections it should be clear that as far as parallel computing is concerned, matrix multiplication is very central problem. Several graph problems may be efficiently solved by a very straightforward application of matrix multiplication. In the examples we have given, the parallel way to solve a graph problem is quite different from the methods used in the best known sequential algorithms. This difference diesn't always show up. For example the algorithm of [4] for the bottleneck (or maximum flow matrix) problem directly translates into an $O(\log^2 n)$ parallel algorithm. Similarly the algorithms obtained in [8] for the shortest cycle problem and the problem of determining if a graph has a triangle directly translate to $O(\log^2 n)$ and $O(\log n)$ matrix multiplication based parallel algorithm. Several other graph problems can be trivially solved in $O(\log^2 n)$ time on a parallel machine using matrix multiplication. We shall refrain from listing these here.

References

1. Agerwala, T. and Lint, B., "Communication in parallel algorithms for Boolean matrix multiplication," Proc. 1978 Int. Conf. on Parallel Processing, IEEE, pp. 146-153, 1978.

2. Cannon L. E., "A cellular computer to implement the Kalman filter algorithm," Ph.D. thesis, Montana State University, 1969.

3. Chandra, A., "Maximal parallelism in matrix multiplication," IBM technical report R C 6193, Sept. 1976.

4. Chen, I-Ngo, "A new parallel algorithm for network flow problems," Proceedings 1974 Sagamore Computer Conference.

5. Dekel, E., Nassimi, D. and Sahni, S., "Parallel matrix and graph algorithm," Univ. of Minnesota, Technical Report #79-10.

6. Flynn, M. J., "Very high speed computing systems," Proc. IEEE 54, Dec. 1966, pp.1901-1909.

7. Hirschberg, D.S., "Parallel algorithms for the transitive closure and the connected component problems," Proc. 8th ACM Symp. on Theo. of Comput., May 1976, pp.55-57.

8. Itai, A. and Rodeh, M., "Finding a minimum circuit in a graph," SICOMP, 7(4), 1978, pp.413-423.

9. Horowitz, E. and Sahni, S., "Fundamentals of Data Structures," Computer Science Press, Potomac, MN, 1976.

10. Pan, V. Ya., "Strassen's algorithm is not optimal. Trilinear technique of aggregating, uniting, and cancelling for constructing fast algorithms for matrix operations," 19th Annual Symposium on Fondations of Computer Science, 1978, pp. 166-176.

11. Nassimi, D. and Sahni, S., "Parallel permutation and sorting algorithms and a new generalized connection network," Univ. of Minnesota, T.R. 79-9, 1979.

PARALLEL ALGORITHMS FOR SOLVING SYSTEMS OF NONLINEAR EQUATIONS

H. MUKAI
Department of Systems Science and Mathematics
Washington University
Box 1040
St. Louis, Missouri 63130

ABSTRACT

In this paper, parallel algorithms are proposed for solving systems of nonlinear algebraic equations. Theoretical results for their convergence and orders of convergence are also presented. The proposed algorithms are analytically compared in terms of their time efficiencies and speed-up ratios.

1. INTRODUCTION

In this paper, two strategies are discussed for improving the efficiency of Newton's methods for solving several nonlinear equations when serial and parallel computers are used. Under some simplifying assumptions, it is found that the strategy of deferred updating is superior to the strategy of partial updating when the degree of parallelism is relatively low, but that the reverse holds when the degree of parallelism is relatively high. To the best of our knowledge, this is the first paper to discuss the efficiency of methods for solving systems of nonlinear algebraic equations in the environment of parallel computers.

2. PROBLEM AND NEWTON'S METHODS

Consider the following nonlinear algebraic vector equation:

$$g(x) = 0, \tag{1}$$

where $g:R^n \to R^n$ is assumed to be continuously differentiable. Let $N = \{1, 2,\dots,n\}$ and let $G(x)$ denote the Jacobian matrix of $g(x)$.

Although many methods have been proposed for locating a solution to (1), Newton's method remains one of the most widely used among them. Given a solution estimate $x_i \in R^n$ Newton's method generates the next estimate x_{i+1} by

$$x_{i+1} = x_i - G(x_i)^{-1} g(x_i). \tag{2}$$

The root-order (the R-order of Ortega and Rheinboldt [8]) of convergence for this method is two. That is, if $x_i \to x^*$ as $i \to \infty$, then

$$||x_i - x^*|| \leq K\theta^{2^i} \text{ for all } i, \tag{3}$$

for some $K > 0$ and $\theta \in (0,1)$.

To avoid the evaluation of the n×n Jacobian matrix $G(x_i)$, the following finite difference approximation J_i to $G(x_i)$ has been proposed. Let e_j be the j-th column of the n×n identity matrix I. Let the j-th column $J_i e_j$ of the matrix J_i be defined by

$$J_i e_j = [g(x_i + \varepsilon_i e_j) - g(x_i)]/\varepsilon_i, \; j \in N, \tag{4}$$

where ε_i satisfies $0 < \varepsilon_i \leq ||g(x_i)||$. Given a solution estimate $x_i \in R^n$, the discrete Newton method generates the next estimate x_{i+1} by

$$x_{i+1} = x_i - J_i^{-1} g(x_i). \tag{5}$$

The root-order of convergence for this method is two, the same as that for Newton's method. (See Traub [12] and Brent [2].) Hence, (3) also holds for the discrete Newton's method. Although this method does not require evaluation of the Jacobian matrix, it requires evaluation of $(n + 1)$ gradients, $g(x_i)$, $g(x_i + \varepsilon_i e_j)$, $j = 1,2,\ldots,n$, at each iteration.

In each of the above methods, much computation time is spent evaluating the current Jacobian or its finite difference approximation. Consequently, for the purpose of speeding up the processes, two strategies have emerged which spread out the computational work of the Jacobian evaluation over several iterations. One strategy may be called partial updating and can be seen in methods proposed by Wolfe [13], Barnes [1] and Polak [10]. The other strategy may be called deferred updating and can be seen in methods presented by Traub [12], Shamanskii [11] and Brent [2]. The first strategy is delineated in the next section and the second, in the section immediately following the next.

3. PARTIAL UPDATING STRATEGY

In the Newton methods reviewed in the preceding section, the whole matrix of the Jacobian or of its approximation is evaluated at each iteration. In the strategy of partial updating, however, only k columns of the matrix are updated and the rest of the columns are retained from the previous iteration. Let $K(i)$ denote the indices of the columns being updated at iteration i. Then the methods based on the partial updating strategy operate as follows. Given a solution estimate $x_i \in R^n$ and Jacobian estimate $J_{i-1} \in R^{n\times n}$, the columns of the new Jacobian estimate J_i are defined by

$$J_i e_j = G(x_i) e_j, \tag{6}$$

or

$$J_i e_j = [g(x_i + \varepsilon_i e_j) - g(x_i)]/\varepsilon_i, \tag{7}$$

for $j \in K(i)$ and

$$J_i e_j = J_{i-1} e_j \tag{8}$$

for $j \in N \sim K(i)$, and the new solution estimate x_{i+1} is defined by (5).

The selection of the k columns to be updated is arbitrary so long as each column is updated at least once in $[n/k]$† iterations. In other words, $K(i)$ is defined so that $K(i)$ contains k indices from N and satisfies

$$N = \bigcup_{m=i-[n/k]+1}^{i} K(m), \quad \forall i \geq [n/k]. \tag{9}$$

The simplest way to define such a function $K(i)$ is

$$K(i) = \{m(i\,k + 1), m(i\,k + 2),\ldots,m(i\,k + k)\},$$

where $m(j)$ denotes the integer in N which is equal to j modulo n.

Because J_i differs from J_{i-1} only in k columns, the inverse J_i^{-1} in (5) may be computed through the following updating formula:

$$J_i^{-1} = J_{i-1}^{-1} + (E - J_{i-1}^{-1} D)(C^T D)^{-1} C^T, \tag{10}$$

where E, D and C are $n\times k$ submatrices respectively made of the k columns of I, J_i and J_{i-1} corresponding to the most recently updated k columns of J_i.

---

† $[n/k]$ denotes the integer m satisfying $m - 1 < n/k \leq m$.

Because this updating formula only requires the inversion of a k×k matrix, C^TD, it generally takes less time than the straightforward inversion of an n×n matrix J_i. However, the formula may be more susceptable to the accumulation of round-off errors.

As stated earlier, the strategy of partial updating can be seen in the so called secant methods proposed by Wolfe [13], Barnes [1] and Polak [10]. However, each of these existing methods updates only one column at each iteration. Our idea of updating several columns at each iteration does not seem to have appeared elsewhere.

Now the methods based on partial updating strategy are put forth as algorithms. The first two algorithms approximate some or all of the columns of the Jacobian matrix by their finite-difference approximations. The two algorithms following the first two employ the exact Jacobian evaluation.

Algorithm 1AS (Approximate version for Serial computers)

Parameters: $k \in N = \{1,2,\ldots,n\}$, $\gamma > 0$.

Step 0: Set $i = 0$. Select an $x_0 \in R^n$, an $\varepsilon_0 > 0$, and an n×n matrix J_0.

Remark: J_0 may be computed through (6) or (7) for $j \in N$.

Step 1: Compute $g(x_0)$. If $g(x_0) = 0$, stop.

Step 2: Compute x_{i+1} through (5).

Remark: Given $g(x_i)$ and J_i, $J_i^{-1}\, g(x_i)$ may be computed from the LU decomposition of J_i or it may be computed from J_i^{-1}, which is computed through (10).

Step 3: Compute $g(x_{i+1})$. If $g(x_{i+1}) = 0$, stop.

Step 4: If $\varepsilon_i \leq \gamma||g(x_{i+1})||$, set $\varepsilon_{i+1} = \varepsilon_i$; else, set $\varepsilon_{i+1} = \gamma||g(x_{i+1})||$.

Remark: The parameter ε_{i+1} thus satisfies

$$\varepsilon_{i+1} \leq \gamma||g(x_{i+1})||. \tag{11}$$

Step 5: Compute either $G(x_{i+1})e_j$, or $g(x_{i+1} - \varepsilon_{i+1}e_j)$ for $j \in K(i)$.

Step 6: Compute J_{i+1} through either (6) or (7) for $j \in K(i)$ and through (8) for $j \in N \sim K(i)$.

Step 7: Set $i = i + 1$. Go to step 2. □

In the above algorithm for serial computers, the parameter ε_{i+1} is maintained in the same order as $g(x_{i+1})$. (See (11)). But in the next algorithm for parallel computers, the parameter ε_{i+1} generally cannot be maintained in the same order as $g(x_{i+1})$, because $g(x_{i+1})$ is evaluated in parallel at the same time as the $g(x_{i+1} - \varepsilon_{i+1}e_j)$ are evaluated; hence the parameter ε_{i+1} is maintained in the same order as $g(x_i)$. As will be seen later, this leads to a slight deterioration in the root order of convergence, but it is amply compensated by the increase in speed due to parallel evaluation. Of course, this deterioration does not occur in the exact version, because it does not involve the parameter ε_{i+1}.

Algorithm 1AP (Approximate version for Parallel computers)

The same as Algorithm 1AS except Steps 3 through 5 are replaced by the following steps:

Step 3: Determine ε_{i+1} by executing either step 3a or 3b.

Step 3a: If $\varepsilon_i \leq \gamma||x_{i+1} - x_i||$, set $\varepsilon_{i+1} = \varepsilon_i$; else, set $\varepsilon_{i+1} = \gamma||x_{i+1} - x_i||$.

Step 3b: If $\varepsilon_i \leq \gamma||g(x_i)||$, set $\varepsilon_{i+1} = \varepsilon_i$; else, set $\varepsilon_{i+1} = \gamma||g(x_i)||$.

Remark: The parameter ε_{i+1} thus satisfies

$$\varepsilon_{i+1} \leq \gamma||g(x_i)|| \text{ or } \varepsilon_{i+1} \leq \gamma||x_{i+1} - x_i||. \tag{12}$$

Step 4: Compute $g(x_{i+1})$ and either $G(x_i)e_j$, or $g(x_{i+1} + \varepsilon_{i+1}e_j)$, $j \in K(i)$.

Step 5: If $g(x_{i+1}) = 0$, stop. ☐

In the above algorithm, there are several places where parallel computation may be employed. In Step 0, the n functions, $G(x_0)e_j$ or $g(x_0 + \varepsilon_0 e_j)$, $j \in N$, may be evaluated in parallel. In Step 2, the LU decomposition of an $n \times n$ matrix may be performed by the existing parallel techniques (see Heller [3] and Miranker [4]), and the algebraic operations necessary for formula (10) are very amenable to parallel computaion. In Step 4, the $(k + 1)$ functions, $g(x_{i+1})$ and either $G(x_i)e_j$ or $g(x_{i+1} + \varepsilon_{i+1}e_j)$, $j \in K(i)$, may be evaluated in parallel. Steps 5 and 6 may be executed in parallel. Furthermore, the computation of columns of J_{i+1} in Step 6 may be divided into several parallel computations.

When the exact Jacobian is easily obtainable, there is no need to approximate it by its finite-difference approximation, and the preceding algorithms become simpler. First, Algorithms 1AS yields the following.

Algorithm 1ES (Exact version for Serial computers)

Parameters: $k \in N = \{1,2,\ldots,n\}$, $\gamma > 0$.

Step 0: Set $i = 0$. Select an $x_0 \in R^n$. Compute $J_0 = G(x_0)$.

Step 1: Compute $g(x_0)$. If $g(x_0) = 0$, stop.

Step 2: Compute x_{i+1} through (5).

Step 3: Compute $g(x_{i+1})$. If $g(x_{i+1}) = 0$, stop.

Step 4: Compute $G(x_{i+1})e_j$, $j \in K(i)$.

Step 5: Compute J_{i+1} through (6) for $j \in K(i)$ and through (8) for $j \in N \sim K(i)$.

Step 6: Set $i = i + 1$. Go to Step 2. ☐

Similarly, Algorithm 1AP yields the following.

Algorithm 1EP (Exact version for Parallel computers)

The same as Algorithm 1ES except Steps 3 and 4 are replaced by the following:

Step 3: Compute $g(x_{i+1})$ and $G(x_i)e_j$, $j \in K(i)$.

Step 4: If $g(x_{i+1}) = 0$, stop. ☐

We shall now establish local convergence and the root-order convergence for Algorithm 1 (i.e., Algorithm 1AS, 1AP, 1ES, and 1EP) under the following:

Assumption 1: A solution x^* to $g(x) = 0$ exists and the following hold.

(i) The Jacobian $G(x^*)$ at x^* is nonsingular.

(ii) The function $g(\cdot)$ is Lipschitz continuously differentiable in a neighborhood of x^*, i.e., there exist $L > 0$ and $r_0 > 0$ such that

$$||G(x') - G(x'')|| \leq L\, ||x' - x''||, \tag{13}$$

$$\forall x', x'' \in B(x^*, r_0) = \{x \mid ||x - x^*|| \leq r_0\}.$$ ☐

With this assumption, the following local convergence theorem holds for Algorithm 1.

Theorem 1: If x_0 and J_0 are sufficiently close to x^* and $G(x_0)$ respectively, then the generated Jacobians or approximates J_i are all invertible and the algorithm either stops at x^* after a finite number of iterations or it generates an infinite sequence $\{x_i\}$ converging to x^*. ☐

See [7] for the proof of this theorem. The following proposition defines the number τ_q as a unique positive root of a certain polynomial.

Proposition 1: For any integer $q \geq 0$, the polynomial $t^{q+1} - t^q - 1$ has a unique positive root τ_q. Moreover, $\tau_q \in (1,2]$, $\tau_q > \tau_{q+1}$ and $\lim_{q\to\infty} \tau_q = 1$. ☐

The proof for this proposition is found in result 9.2.8 in Ortega and Rheinboldt [8].

Recall that if an iterative process with root-order τ generates a sequence $\{x_i\}$ converging to x^* then there exist $K > 0$ and $\theta \in (0,1)$ such that

$$||x_i - x^*|| \leq K\,\theta^{(\tau)^i}, \quad i = 0, 1, 2, \ldots \tag{14}$$

The following two theorems establish the orders of convergence for Algorithms 1AS, 1AP, 1ES and 1EP. Their proofs are found in [7].

Theorem 2: The root-orders of Algorithms 1AS, 1ES and 1EP are at least $\tau_{[n/k]-1}$. ☐

Theorem 3: The root-order of Algorithm 1AP is at least $\tau_{[n/k\rfloor}$. ☐

4. DEFERRED UPDATING STRATEGY

Seeking a higher order, Traub [12] studied the following variation of Newton's method, in which the Jacobian is only occasionally reevaluated. Given a solution estimate $x_i \in R^n$, the next estimate x_{i+1} is computed as a result of k Newton iterations with the fixed Jacobian $G(x_i)$, i.e. define

$$x_{i+1} = y_{i,k}, \tag{15}$$

where $y_{i,0}, y_{i,1}, \ldots, y_{i,k}$ are generated by

$$y_{i,0} = x_i, \tag{16}$$

and

$$y_{i,\ell} = y_{i,\ell-1} - G(x_i)^{-1}\, g(y_{i,\ell-1}), \text{ for } \ell = 1,2,\ldots,k. \tag{17}$$

Next, Shamanskii [11] proposed a discrete version of Traub's method, in which $G(x_i)$ in (17) is replaced by its approximate J_i computed through the finite difference approximation (4) with ε_i satisfying $0 < \varepsilon_i \leq ||g(x_i)||$.

These methods are now presented in the form of algorithms. The first algorithm approximates some or all of the columns of the Jacobian matrix, while the second and third evaluate the exact Jacobian matrix.

Algorithm 2A (Approximate version)

Parameters: $k \in Z = \{1,2,\ldots\}$, $\gamma > 0$.

Step 0: Set $i = 0$ and select an $\varepsilon_0 > 0$.

Step 1: Compute $g(x_i)$. If $g(x_i) = 0$, stop.

Step 2: If $\varepsilon_i > \gamma\, ||g(x_i)||$, set $\varepsilon_i = \gamma\, ||g(x_i)||$.

Step 3: Evaluate $G(x_i)e_j$ or $g(x_i + \varepsilon_i e_j)$, $j \in N$.

Step 4: Compute J_i through (6) or (7) for $j \in N$.

Step 5: Obtain the LU decomposition of J_i.

Step 6: Set $\ell = 0$ and $y_{i,0} = x_i$.

Step 7: If $\ell > 0$, compute $g(y_{i,\ell})$.

Remark: When $\ell = 0$, $g(y_{i,0}) = g(x_i)$ has been already evaluated in step 1.

Step 8: Compute

$$y_{i,\ell+1} = y_{i,\ell} - J_i^{-1} g(y_{i,\ell}) \tag{18}$$

Step 9: Set $\ell = \ell + 1$.

```
Step 10:    If $\ell < k$, go to step 7; else, go to step 11.
Step 11:    Set $x_{i+1} = y_{i,k}$, set $\varepsilon_{i+1} = \varepsilon_i$, set $i = i + 1$ and go to step 1.    □
```

When the exact Jacobian is easily obtainable, there is no need to employ the finite difference approximation for the Jacobian and the algorithm becomes simpler.

Algorithm 2ES (Exact version for Serial computers)

```
Parameters: $k \in Z = \{1,2,\dots\}$, $\gamma > 0$.
Step 0:     Set $i = 0$ and select an $\varepsilon_0 > 0$.
Step 1:     Compute $g(x_i)$. If $g(x_i) = 0$, stop.
Step 2:     Compute $J_i = G(x_i)$.
Step 3:     Obtain the LU decomposition of $J_i$.
Step 4:     Set $\ell = 0$ and $y_{i,0} = x_i$.
Step 5:     If $\ell > 0$ then compute $g(y_{i,\ell})$.
Step 6:     Compute $y_{i,\ell+1}$ through (18).
Step 7:     Set $\ell = \ell + 1$.
Step 8:     If $\ell < k$, go to step 5; else go to step 9.
Step 9:     Set $x_{i+1} = y_{i,k}$, set $i = i + 1$ and go to step 1.    □
```

When a parallel computer is available, steps 1 and 2 in Algorithms 2ES may be executed in parallel. This yields the following algorithm.

Algorithm 2EP (Exact version for Parallel computers)

The same as Algorithm 2ES except its steps 1 and 2 are replaced by the following.

```
Step 1:     Evaluate $g(x_i)$ and $J_i = G(x_i)$.
Step 2:     If $g(x_i) = 0$, stop.    □
```

Algorithms 2ES and 2EP become the classic Newton's method, when $k = 1$.

The local convergence (Theorem 1) and the root-order of convergence of at least $(k + 1)$ have been established by Traub [12] for Algorithm 2ES and 2EP and by Shamanskii [11] for Algorithm 2A. (See also discussions NR10.2-2 and NR11.2-5 in Ortega and Rheinboldt [8])

5. EFFICIENCY ANALYSIS IN SERIAL COMPUTATION

In order to compare speeds of different iterative methods, Brent [2] defined the (work) efficiency E of an iterative process with a superlinear convergence by

$$E = (\log \rho)/W \tag{19}$$

where

$$\rho = \liminf_{i\to\infty} (-\log ||x_i - x^*||)^{1/i} \tag{20}$$

is the root-order of convergence of the iterative method, and

$$W = \lim_{i\to\infty} W_i \tag{21}$$

is the limit of the amount W_i of work required to compute x_i at iteration i.

Because time is a more appropriate measure than work in the context of parallel computation, we modify (19) a little and define the time efficiency of an iterative method with a superlinear convergence by

$$E = (\log \rho)/T \tag{22}$$

where

$$T = \lim_{i\to\infty} T_i \tag{23}$$

is the limit of the amount T_i of time required to compute x_i at iteration i.

With this definition, a method with time efficiency E requires E'/E times as much time as a method with time efficiency E' to reduce the error $||x_i - x^*||$ to a very small positive tolerance.

In this section, time efficiencies are discussed for the methods presented in the proceding sections when they are run on serial computers.

The computational time T_i at iteration i may be measured in any appropriate units; however, we adopt as a time unit the length of time required for one vector function evaluation, i.e., the evaluation of g(x), given x. In order to simplify the analysis, the following assumption is made.

Assumption 2: (i) The evaluation of any column of G(x) takes the same amount of time as that of evaluation of any g(x).

(ii) The evaluation of the function g(x) (and possibly of columns of its Jacobian G(x)) dominates the computation time. □

From this point on, the evaluation of one column of G(x) is considered one function evaluation just as the evaluation of g(x).

First, consider Algorithms 1AS and 1ES, for which the root-order ρ is at least $\tau_{[n/k]-1}$. Because they require (k + 1) function evaluations at each iteration, the time efficiency for them is at least

$$E_{1S} = (\log \tau_{[n/k]-1})/(k + 1) \tag{24}$$

As noted before, Algorithms 1AS and 1ES subsume Newton's method when k = n and the secant method when k = 1. Indeed, the time efficiencies for these methods are respectively,

$$E_N = (\log \tau_0)/(n + 1), \quad E_S = (\log \tau_{n-1})/2 \tag{25/26}$$

Given n∈Z, the optimal parameter k^*_{1S} among N = {1,2,...,n} may be selected as the k that maximizes (24). This algorithm optimization will be called optimal tuning. The time efficiency of the resulting optimally tuned Algorithms 1AS and 1ES is at least

$$E^*_{1S} = (\log \tau_{[n/k^*_{1S}]-1})/(k^*_{1S} + 1) \tag{27}$$

Some sample values for k^*_{1S} and E^*_{1S} are found in Table 1.

By definition, the optimally tuned Algorithms 1AS and 1ES are generally more time-efficient than both the Newton's and the secant methods. Compare the time efficiencies of these methods in Table 1.

Consider Algorithms 2A and 2ES, for which the root-order ρ is at least (k + 1). Since they require (n + k) function evaluations at each iteration, the time efficiency for each is at least

$$E_{2S} = (\log (k + 1))/(n + k) \tag{28}$$

Again, a maximizer k^*_{2S} of (28) may be selected and, with such a k, (28) becomes

$$E^*_{2S} = (\log (k^*_{2S} + 1))/(n + k^*_{2S}). \tag{29}$$

Since Algorithms 2A and 2ES subsume Newton's method (when k = 1), the optimally tuned Algorithms 2A and 2ES are more efficient than Newton's method. See some samples of E_N, E^*_{2S} and k^*_{2S} in Table 1.

Comparing E^*_{1S} and E^*_{2S} in Table 1, we conclude that the optimally tuned Algorithms 2A and 2ES are more time-efficient than the optimally tuned Algorithms 1AS and 1ES.

6. EFFICIENCY ANALYSIS IN PARALLEL COMPUTATION

In this section time efficiencies are discussed for the methods presented in the preceding sections when they are run on parallel computers with p parallel processors.

In comparing different parallel algorithms, the so called speed-up ratio and (parallel) efficiency are useful measures. The speed-up ratio measures how many times faster a parallel algorithm on a parallel computer is as compared with a serial algorithm on a serial computer for the same problem. In other words, the speed-up ratio is defined to be a ratio of the computation time on a serial computer to the computation time on a parallel computer. The parallel efficiency of a method is defined to be its speed-up ratio divided by the number of parallel processors used in the computation, and it measures how efficient the method is in its utilization of parallel processors.

Assumption 2 is still in effect in this section. If the task of evaluating one function (either $g(x)$ or a column of $G(x)$) is divided among several processors, then the associated programming becomes more complicated and costly, and, furthermore, the speed of the algorithm may not increase appreciably because of higher overhead due to increased coordination requirement. Hence, the following assumption is made.

Assumption 3: The evaluation of one function ($g(x)$ or a column of $G(x)$) is entirely performed by one processor. □

Consider Algorithm 1AP, for which the root-order ρ is at least $\tau_{[n/k]}$, and Algorithm 1EP, for which the root-order ρ is at least $\tau_{[n/k]-1}$. With p processors, the parallel evaluation of $(k + 1)$ functions found in step 4 of Algorithm 1AP and in step 3 of Algorithm 1EP takes $[(k + 1)/p]$ units of time, where one unit of time is the time required for one function evaluation by one processor. Hence, the time efficiencies for these algorithms are respectively at least

$$E_{1AP} = (\log \tau_{[n/k]})/[(k + 1)/p] \tag{30}$$

$$E_{1EP} = (\log \tau_{[n/k]-1})/[(k + 1)/p] \tag{31}$$

Given n and p in Z, it is easy to determine the values k^{**}_{1AP} and k^{**}_{1EP} that respectively maximize (30) and (31) over $k \in \{1,2,\ldots,n\}$. The resulting optimally tuned Algorithms 1AP and 1EP have respective time efficiencies of at least

$$E^{**}_{1AP} = (\log \tau_{[n/k^{**}_{1AP}]})/[(k^{**}_{1AP} + 1)/p] \tag{32}$$

$$E^{**}_{1EP} = (\log \tau_{[n/k^{**}_{1EP}]-1})/[(k^{**}_{1EP} + 1)/p] \tag{33}$$

When compared with their respective serial versions (the optimally tuned Algorithms 1AS and 1ES), the speed-up ratios and the parallel efficiencies (see Table 2) are respectively

$$S_{1AP} = E^{**}_{1AP}/E^{*}_{1S} \qquad S_{1EP} = E^{**}_{1EP}/E^{*}_{1S} \tag{34/35}$$

$$\mathcal{E}_{1AP} = S_{1AP}/p \qquad \mathcal{E}_{1EP} = S_{1EP}/p \tag{36/37}$$

Consider Algorithm 2A, for which the root-order is at least $(k + 1)$. With p processors, the evaluation of n functions at step 3 takes $[n/p]$ units of time. The evaluation of $g(x_i)$ at step 1 and the $(k + 1)$ sequential evaluations of $g(y_{i,\ell})$ at step 8 take k units of time. Hence, the time efficiency for Algorithm 2A is at least

$$E_{2A} = (\log(k + 1))/([n/p] + k) \tag{38}$$

Next, consider Algorithm 2EP, for which the root-order is at least $(k + 1)$. With p processors, the evaluation of $(n + 1)$ functions at step 1 takes $[(n + 1)/p]$ units of time and the $(k + 1)$ sequential evaluations of $g(y_{i,\ell})$ at step 5 take $(k - 1)$ units of time. Hence, the time efficiency is at least

$$E_{2EP} = (\log(k + 1))/([(n+1)/p] + k - 1) \tag{39}$$

Given n and p in Z, let k^{**}_{2A} and k^{**}_{2EP} denote the values of k that respectively maximize (38) and (39) over positive integers. The resulting optimally tuned Algorithms 2A and 2EP yield time efficiencies of at least

$$E^{**}_{2A} = (\log(k^{**}_{2A} + 1))/([n/p] + k^{**}_{2A}) \qquad (40)$$

$$E^{**}_{2EP} = (\log(k^{**}_{2EP} + 1))/([(n + 1)/p] + k^{**}_{2EP} - 1) \qquad (41)$$

When compared with its serial versions (the optimally tuned Algorithms 2AS and 2ES), the speed-up ratios and parallel efficiencies (see Table 3) of the optimally tuned Algorithms 2A and 2EP are given by

$$S_{2A} = E^{**}_{2A}/E^{*}_{2S} \qquad S_{2EP} = E^{**}_{2EP}/E^{*}_{2S} \qquad (42/43)$$

$$\mathcal{E}_{2A} = S_{2A}/p \qquad \mathcal{E}_{2EP} = S_{2EP}/p \qquad (44/45)$$

Finally, the time efficiencies of the four parallel methods are compared in Table 4. First, compare the two approximate algorithms and observe that Algorithm 2A is more time-efficient than Algorithm 1AP when the number p of processors is low, but that the reverse is true, when p is high. Next, compare the two exact algorithms and note that when p = n + 1, their efficiencies are the same because they become identical. Observe further that Algorithm 2EP is more time-efficient than Algorithm 1EP, when the number p of processors is low, but that the reverse is true, when p is high.

8. REFERENCES

[1] J. Barnes, "An algorithm for solving nonlinear equations based on the secant method," Computer Journal, 8 (1965) 66-72.

[2] R. P. Brent, "Some efficient algorithms for solving systems of nonlinear equations," SIAM J. on Numerical Analysis 10 (1973) 327-344.

[3] D. Heller, "A survey of parallel algorithms in numerical linear algebra," SIAM Review, 20 (1978) 740-777.

[4] W. L. Miranker, "A survey of parallelism in numerical analysis," SIAM Review, 13 (1971) 524-547.

[5] W. L. Miranker, "Parallel methods for solving equations," Parallel Computers - Parallel Mathematics, Ed. by M. Feilmeier, (1977) 9-15, also Mathematics and Computers in Simulation 20 (1978) 93-101.

[6] H. Mukai, "Parallel algorithms for unconstrained optimization," to appear in the Proceeding of the IEEE Conference on Decision and Control, Fort Lauderdale, Florida, (1979).

[7] H. Mukai, "Parallel algorithms for solving systems of nonlinear equations," SSM Report No. 7914, Dept. of Systems Science and Mathematics, Washington University, St. Louis, MO, (1979).

[8] J. M. Ortega and W. C. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press (1970).

[9] A. M. Ostrowski, Solution of Equations and Systems of Equations, Academic Press (1960).

[10] E. Polak, "A globally converging secant method with applications to boundary value problems," SIAM Journal on Numerical Analysis 11 (1974) 529-537.

[11] V. E. Shamanskii, "On a modification of Newton's method," Ukrain. Mat. Z. 19 (1967) 133-138.

[12] J. Traub, Iterative Methods for the Solution of Equations, Prentice Hall, Englewood Cliffs, New Jersey (1964).

[13] P. Wolfe, "The secant method for simultaneous nonlinear equations," Communications of the ACM 2 (1959) 12-14.

Table 1. Methods for Serial Computation

Dimen-sion	Optimal Parameter		Time Efficiencies			
	Algo. 1S	Algo. 2S	Secant	Algo. 1S	Newton	Algo. 2S
n	k^*_{1S}	k^*_{2S}	E_S	E^*_{1S}	E_N	E^*_{2S}
1	1	2	0.3466	0.3466	0.3466	0.3662
2	1	3	0.2406	0.2406	0.2310	0.2773
5	1	5	0.1406	0.1406	0.1155	0.1792
10	2	7	0.0901	0.0937	0.0630	0.1223
20	2	11	0.0561	0.0601	0.0330	0.0802
50	2	22	0.0288	0.0319	0.0136	0.0435
100	2	37	0.0170	0.0192	0.0069	0.0266
200	3	62	0.0098	0.0116	0.0034	0.0158
500	4	128	0.0047	0.0057	0.0014	0.0077
1000	4	225	0.0026	0.0033	0.0007	0.0044
2000	5	399	0.0015	0.0019	0.0003	0.0025
5000	5	866	0.0007	0.0009	0.0001	0.0012

Table 2. Partial Updating Methods for Parallel Computation

Dimen-sion	Number of processors	Optimal Parameters		Time Efficiencies		Speed-up Ratios		Parallel Efficiencies	
n	p	k^{**}_{1AP}	k^{**}_{1EP}	E^{**}_{1AP}	E^{**}_{1EP}	S_{1AP}	S_{1EP}	E_{1AP}	E_{1EP}
10	1	1	2	0.0846	0.0937	0.90	1.00	0.90	1.00
	2	1	1	0.1691	0.1802	0.81	1.92	0.90	0.96
	5	4	4	0.3223	0.3823	3.44	4.08	0.69	0.82
	6	5	5	0.3822	0.4812	4.08	5.14	0.68	0.86
	10	5	5	0.3822	0.4812	4.08	5.14	0.41	0.51
	11	10	10	0.4812	0.6931	5.14	7.40	0.47	0.67
100	1	2	2	0.0189	0.0192	0.98	1.00	0.98	1.00
	2	3	3	0.0375	0.0383	1.95	1.99	0.98	1.00
	5	4	4	0.0931	0.0957	4.85	4.98	0.97	1.00
	6	5	5	0.1084	0.1122	5.65	5.84	0.94	0.97
	10	9	9	0.1511	0.1595	7.87	8.31	0.79	0.83
	11	10	10	0.1691	0.1802	8.81	9.39	0.80	0.85
	50	34	34	0.3223	0.3822	16.79	19.91	0.34	0.40
	51	50	50	0.3822	0.4812	25.06	25.06	0.49	0.49
	100	50	50	0.3822	0.4812	25.06	25.06	0.25	0.25
	101	100	100	0.4812	0.6931	36.10	31.10	0.36	0.36
1000	1	4	4	0.0033	0.0033	1.00	1.00	1.00	1.00
	2	5	5	0.0065	0.0066	1.97	2.00	0.98	1.00
	5	4	4	0.0164	0.0165	4.97	5.00	0.99	1.00
	6	5	5	0.0196	0.0197	5.94	5.97	0.99	0.99
	10	9	9	0.0309	0.0311	9.36	9.42	0.94	0.94
	11	10	10	0.0337	0.0340	10.21	10.30	0.93	0.94
	50	48	48	0.1049	0.1084	31.79	32.85	0.64	0.66
	51	50	50	0.1084	0.1122	34.00	34.00	0.67	0.67
	100	91	91	0.1595	0.1691	48.33	51.24	0.48	0.51
	101	100	100	0.1691	0.1802	51.24	54.61	0.51	0.54
	500	334	334	0.3223	0.3822	97.67	115.82	0.20	0.23
	501	500	500	0.3822	0.4812	115.82	145.82	0.23	0.29
	1000	500	500	0.3822	0.4812	115.82	145.82	0.12	0.15
	1001	1000	1000	0.4812	0.6931	145.82	210.03	0.15	0.21

Table 3. Deferred Updating Methods for Parallel Computation

Dimen-sion	Number of Processors	Optimal Parameters		Time Efficiencies		Speed-up Ratios		Parallel Efficiencies	
n	p	k^{**}_{2A}	k^{**}_{2EP}	E^{**}_{2A}	E^{**}_{2EP}	S_{2A}	S_{2EP}	E_{2A}	E_{2EP}
10	1	7	7	0.1223	0.1223	1.00	1.00	1.00	1.00
	2	5	5	0.1792	0.1792	1.47	1.47	0.73	0.73
	5	3	3	0.2773	0.2773	2.27	2.27	0.45	0.45
	6	3	2	0.2773	0.3662	2.27	2.99	0.38	0.50
	10	2	2	0.3662	0.3662	2.99	2.99	0.30	0.30
	11	2	1	0.3662	0.6931	2.99	5.67	0.27	0.52
100	1	37	37	0.0266	0.0266	1.00	1.00	1.00	1.00
	2	22	22	0.0435	0.0435	1.64	1.64	0.82	0.82
	5	11	11	0.0802	0.0802	3.02	3.02	0.60	0.60
	6	10	10	0.0888	0.0922	3.34	3.47	0.56	0.58
	10	7	7	0.1223	0.1223	4.60	4.60	0.46	0.46
	11	7	7	0.1223	0.1300	4.60	4.89	0.42	0.44
	50	3	3	0.2773	0.2773	10.42	10.42	0.21	0.21
	51	3	2	0.2773	0.3662	10.42	13.77	0.20	0.27
	100	2	2	0.3662	0.3662	13.77	13.77	0.14	0.14
	101	2	1	0.3662	0.6931	13.77	26.06	0.14	0.26
1000	1	225	225	0.0044	0.0044	1.00	1.00	1.00	1.00
	2	128	128	0.0077	0.0077	1.75	1.75	0.88	0.88
	5	62	62	0.0158	0.0158	3.59	3.59	0.72	0.72
	6	54	54	0.0181	0.0182	4.11	4.14	0.69	0.69
	10	37	37	0.0266	0.0266	6.05	6.05	0.60	0.60
	11	34	34	0.0284	0.0287	6.45	6.52	0.59	0.59
	50	11	11	0.0802	0.0802	18.23	18.23	0.36	0.36
	51	11	11	0.0802	0.0828	18.23	18.82	0.36	0.37
	100	7	7	0.1223	0.1223	27.80	27.80	0.28	0.29
	101	7	7	0.1223	0.1300	27.80	29.55	0.28	0.29
	500	3	3	0.2773	0.2773	63.02	63.02	0.13	0.13
	501	3	2	0.2773	0.3662	63.02	83.23	0.13	0.17
	1000	2	2	0.3662	0.3662	83.23	83.23	0.08	0.08
	1001	2	1	0.3662	0.6931	83.23	157.52	0.08	0.16

Table 4. Methods for Parallel Computation

Dimensions	Number of Processors	Time Efficiencies			
		Approximate Versions		Exact Versions	
n	p	E^*_{1AP}	E^{**}_{2A}	E^*_{1EP}	E^{**}_{2EP}
10	1	0.0846	0.1223	0.0937	0.1223
	2	0.1691	0.1792	0.1802	0.1792
	5	0.3223	0.2773	0.3823	0.2773
	6	0.3822	0.2773	0.4812	0.3662
	10	0.3822	0.3662	0.4812	0.3662
	11	0.4812	0.3662	0.6931	0.6931
100	1	0.0189	0.0266	0.0192	0.0266
	2	0.0375	0.0435	0.0383	0.0435
	5	0.0931	0.0802	0.0957	0.0802
	6	0.1084	0.0888	0.1122	0.0922
	10	0.1511	0.1223	0.1595	0.1223
	11	0.1691	0.1223	0.1802	0.1300
	50	0.3223	0.2773	0.3822	0.2773
	51	0.3822	0.2733	0.4812	0.3662
	100	0.3822	0.3662	0.4812	0.3662
	101	0.4812	0.3662	0.6931	0.6931
1000	1	0.0033	0.0044	0.0033	0.0044
	2	0.0065	0.0077	0.0066	0.0077
	5	0.0164	0.0158	0.0165	0.0158
	6	0.0196	0.0181	0.0197	0.0182
	10	0.0309	0.0266	0.0311	0.0266
	11	0.0337	0.0284	0.0340	0.0287
	50	0.1049	0.0802	0.1084	0.0802
	51	0.1084	0.0802	0.1122	0.0828
	100	0.1595	0.1223	0.1691	0.1223
	101	0.1691	0.1223	0.1802	0.1300
	500	0.3223	0.2773	0.3822	0.2773
	501	0.3822	0.2773	0.4812	0.3662
	1000	0.3822	0.3662	0.4812	0.3662
	1001	0.4812	0.3662	0.6931	0.6931

BROADCASTING A SMALL NUMBER OF MESSAGES IN A SQUARE GRID GRAPH

FRANCES L. VAN SCOY
West Virginia University
Morgantown, West Virginia 26506

ABSTRACT

Processors are arranged in a square grid with each processor connected to its four nearest neighbors. Initially processor (1,1) has k messages which are to be sent to every other processor in the grid. Each processor is restricted to receiving or transmitting only one message at one time. An algorithm for broadcasting k messages in an n x n grid in time 2n + 2k - 4 when $k \leq n-2$ for odd n and $k \leq n-1$ for even n is presented.

DEFINITIONS

Let G(n) be the square grid graph consisting of the n^2 points labeled by all ordered pairs of integers between 1 and n with an edge between each pair of points (i,j) and (i,j+1) and each pair (i,j) and (i+1,j).

The term broadcasting describes the process by which one or more messages stored at point (1,1) are sent to every other point in the grid graph. The following restrictions hold:

(1) initially point (1,1) contains k messages;

(2) a message may be sent only between points connected by an edge;

(3) time is discrete, and transmission of one message between adjacent points requires only one unit of time;

(4) at one time, a point may either send or receive only one message.

The minimum broadcast time of the square grid graph G(n) is the smallest amount of time required to broadcast k messages throughout the entire grid. This is denoted by T(n,k).

PREVIOUS WORK

Others have studied broadcasting in more general graphs. Work on broadcasting in graphs has been done by Hedetniemi and Mitchell (9), Farley, Hedetniemi, Mitchell, and Proskurowski (8), and Farley (4,5,6). Slater, Cockayne, and Hedetniemi (12) and Proskurowski (11) have examined broadcasting in trees.

Farley and Hedetniemi (7), Cockayne and Hedetniemi (2), and Peck (10) have studied broadcasting in n-dimensional grid graphs. In particular, Cockayne and Hedetniemi (2) obtained a lower bound for g(n,t), the maximum number of vertices of an n-dimensional grid graph which may receive a single message from the originator in t time intervals.

Chinn, Hedetniemi, and Mitchell (1) and Cockayne and Thomason (3) have examined the problem of broadcasting multiple messages in arbitrary graphs.

BROADCASTING ONE MESSAGE IN A GRID GRAPH

Farley and Hedetniemi (7) showed that for the two-dimensional case T(n,1) = 2n - 2 and presented such an algorithm. Their algorithm requires that the message be sent from processor (1,1) across row 1 of the grid as rapidly as possible (in time n-1) and then in parallel down each column from row 1. The movement of the message through the grid is shown in Figure 1 for n = 4, where an arrow indicates the direction of movement of the message and the number on an arrow tells the time at which this movement occurs.

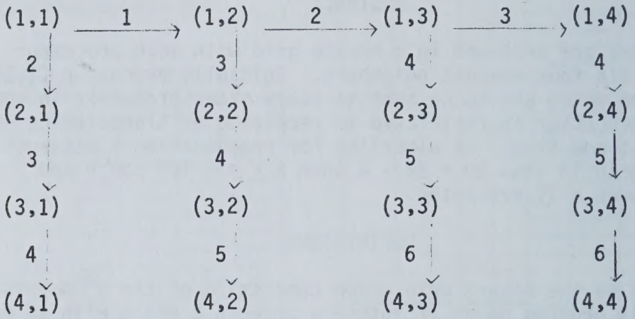


Figure 1
Broadcasting One Message in G(4)

BROADCASTING MULTIPLE MESSAGES IN A GRID GRAPH

To broadcast k messages, this algorithm can be extended by sending the second message and later messages along the same path as the first after a delay of 3 time intervals. The extended algorithm requires time 2n + 3k - 5. When n = 4 and k = 2, message movement occurs at the times indicated in Figure 2.

(1,1)	1,4	(1,2)	2,5	(1,3)	3,6	(1,4)
2,5		3,6		4,7		4,7
(2,1)		(2,2)		(2,3)		(2,4)
3,6		4,7		5,8		5,8
(3,1)		(3,2)		(3,3)		(3,4)
4,7		5,8		6,9		6,9
(4,1)		(4,2)		(4,3)		(4,4)

Figure 2
Broadcasting Two Messages in G(4)

Description of Algorithm

However, $T(n,k) \neq 2n + 3k - 5$. In the remainder of this paper an algorithm is shown which requires time 2n + 2k - 4 to broadcast k messages in a grid graph of size n, when $k \leq n-2$ for odd n and $k \leq n-1$ for even n. The method requires that the comb-like data flow path of the previous algorithm and its 90° rotation be alternated for successive messages. Odd numbered messages are sent from (1,1) to (1,2) at times 1, 5, 9, ... Even numbered messages are sent from (1,1) to (2,1) at times 3, 7, 11, ... Odd numbered messages follow the comb path with parallel vertical paths, while even numbered messages follow the comb path with parallel horizontal paths. The least obvious part of the algorithm is the determination of times at which even numbered messages move along row 1 and odd numbered messages move along column 1. The algorithm is shown for broadcasting 3 messages in a grid with 4 rows and columns in Figure 3. The pair (m)t indicates that message m is to be sent along this edge at time t.

(1,1)	(1)1,(3)5 (2)4	(1,2)	(1)2,(3)6 (2)9	(1,3)	(1)3,(3)7 (2)10	(1,4)
(2)3 (1)2 (3)6		(1)3 (3)7		(1)4 (3)8		(1)4 (3)8
(2,1)	(2)5	(2,2)	(2)6	(2,3)	(2)7	(2,4)
(2)4 (1)7 (3)9		(1)4 (3)8		(1)5 (3)9		(1)5 (3)9
(3,1)	(2)6	(3,2)	(2)7	(3,3)	(2)8	(3,4)
(2)5 (1)8 (3)10		(1)5 (3)9		(1)6 (3)10		(1)6 (3)10
(4,1)	(2)6	(4,2)	(2)7	(4,3)	(2)8	(4,4)

Figure 3
Broadcasting Three Messages in G(4)

The details of timing are shown below. In the statements which follow, $1 < i < n$, $1 < j < n$, odd = the number of odd numbered messages = $\frac{1}{2}k$ if 2 divides k and $\frac{1}{2}(k+1)$ if 2 does not divide k, and even = the number of even numbered messages = $\frac{1}{2}k$ if 2 divides k and $\frac{1}{2}(k-1)$ if 2 does not divide k.

1. Cell (1,1) sends message m to cell (1,2) at time $2m - 1$, when m is odd.

Cell (1,1) sends message m to cell (2,1) at time $2m - 1$, when m is even.

Cell (1,1) sends message m to cell (1,2) at time $2m$, when m is even.

Cell (1,1) sends message m to cell (2,1) at time $2m$, when m is odd.

2. Cell (1,j) receives message m from cell (1,j-1) at time $2m + j - 3$ when m is odd.

Cell (1,j) sends message m to cell (1,j+1) at time $2m + j - 2$ when m is odd.

Cell (1,j) sends message m to cell (2,j) at time $2m + j - 1$ when m is odd.

Cell (1,j) receives message m from cell (1,j-1) at time $2k + n + j - 2(\text{even}) - 4 + m$ when m is even unless $j = 2$.

Cell (1,2) receives message m from cell (1,1) at time $2m$ when m is even.

Cell (1,j) sends message m to cell (i,j+1) at time $2k + n + j - 2(\text{even}) - 3 + m$ when m is even.

3. Cell $(1,n)$ receives message m from cell $(1,n-1)$ at time $2m + n - 3$ when m is odd.

Cell $(1,n)$ sends message m to cell $(2,n)$ at time $2m + n - 2$ when m is odd.

Cell $(1,n)$ receives message m from cell $(1,n-1)$ at time $2k + 2n - 2(\text{even}) - 4 + m$ when m is even.

4. Cell $(i,1)$ receives message m from cell $(i-1,1)$ at time $2m + i - 3$ when m is even.

Cell $(i,1)$ sends message m to cell $(i+1,1)$ at time $2m + i - 2$ when m is even.

Cell $(i,1)$ sends message m to cell $(i,2)$ at time $2m + i - 1$, when m is even.

Cell $(i,1)$ receives message m from cell $(i-1,1)$ at time $2k + n + i - 2(\text{odd}) - 3 + m$ when m is odd unless $i = 2$.

Cell $(2,1)$ receives message m from cell $(1,1)$ at time $2m$ when m is odd.

Cell $(i,1)$ sends message m to cell $(i+1,1)$ at time $2k + n + i - 2(\text{odd}) - 2 + m$ when m is odd.

5. Cell (i,j) receives message m from cell $(i-1,j)$ at time $2m + i + j -3$ when m is odd.

Cell (i,j) sends message m to cell $(i+1,j)$ at time $2m + i + j - 2$ when m is odd.

Cell (i,j) receives message m from cell $(i,j-1)$ at time $2m + i + j -3$ when m is even.

Cell (i,j) sends message m to cell $(i,j+1)$ at time $2m + i + j - 2$ when m is even.

6. Cell (i,n) receives message m from cell $(i-1,n)$ at time $2m + n + i -4$ when m is odd.

Cell (i,n) sends message m to cell $(i+1,n)$ at time $2m + n + i - 3$ when m is odd.

Cell (i,n) receives message m from cell $(i,n-1)$ at time $2m + n + i -3$ when m is even.

7. Cell $(n,1)$ receives message m from cell $(n-1,1)$ at time $2m + n - 3$ when m is even.

Cell $(n,1)$ sends message m to cell $(n,2)$ at time $2m + n - 2$ when m is even.

Cell $(n,1)$ receives message m from cell $(n-1,1)$ at time $2k + 2n - 2(\text{odd}) - 3 + m$ when m is odd.

8. Cell (n,j) receives message m from cell $(n-1,j)$ at time $2m + n + j -3$ when m is odd.

Cell (n,j) receives message m from cell $(n,j-1)$ at time $2m + n + j -4$ when m is even.

Cell (n,j) sends message m to cell $(n,j+1)$ at time $2m + n + j - 3$ when m is even.

9. Cell (n,n) receives message m from cell $(n-1,n)$ at time $2m +2n - 4$ when m is odd.

Cell (n,n) receives message m from cell $(n,n-1)$ at time $2m + 2n - 4$ when m is even.

Proof of Correctness of Algorithm

Examination of statements 1 through 9 shows that the proof of correctness of the algorithm depends on showing that in each cell the times for receiving and sending messages are distinct.

1. Cell (1,1) obvious

2. Cell (1,j)
It suffices to show that, for all odd m' and even m'',
$2m' + j - 1 < 2k + n + j - 2(\text{even}) - 4 + m''$.
Case 1. 2 divides k
$2m' + j - 1 \leq 2(k-1) + j - 1 = 2k + j - 3$
$2k + n + j - 2(\text{even}) + 4 + m'' \geq 2k + n + j - 2(\tfrac{1}{2}k) - 4 + 2$
For $2m' + j - 1 < 2k + n + j - 2(\text{even}) - 4 + m$ it is sufficient that $k \leq n$.
Case 2. 2 does not divide k
$2m' + j - 1 \leq 2k + j - 1$
$2k + n + j - 2(\text{even}) - 4 + m'' \geq k + n + j$
It is sufficient that $k \leq n$.

3. Cell (1,n) special case of cell (1,j)

4. Cell (i,1)
It suffices to show that, for all even m' and odd m'',
$2m' + i - 1 < 2k + n + i - 2(\text{odd}) - 3 + m''$.
Case 1. 2 divides k
$2m' + i - 1 \leq 2k + i - 1$
$2k + n + i - 2(\text{odd}) - 3 + m'' \geq k + n + i - 2$
It is sufficient that $k < n-1$.
Case 2. 2 does not divide k
$2m' + i - 1 \leq 2k + i - 3$
$2k + n + i - 2(\text{odd}) - 3 + m'' \geq k + n + i - 3$
It is sufficient that $k < n$.

5. Cell (i,j) obvious

6. Cell (i,n) obvious

7. Cell (n,1)
It suffices to show that, for all even m' and odd m'',
$2m' + n - 2 < 2k + 2n - 2(\text{odd}) + 3 + m''$.
Case 1. 2 divides k
$2m' + n - 2 \leq 2k + n - 2$
$2k + 2n - 2(\text{odd}) + 3 + m'' \geq k + 2n + 4$
It is sufficient that $k < n + 6$.
Case 2. 2 does not divide k
$2m' + n - 2 \leq 2k + n - 4$
$2k + 2n - 2(\text{odd}) + 3 + m'' \geq k + 2n + 3$
It is sufficient that $k < n + 7$.

8. Cell (n,j) obvious

9. Cell (n,n) obvious

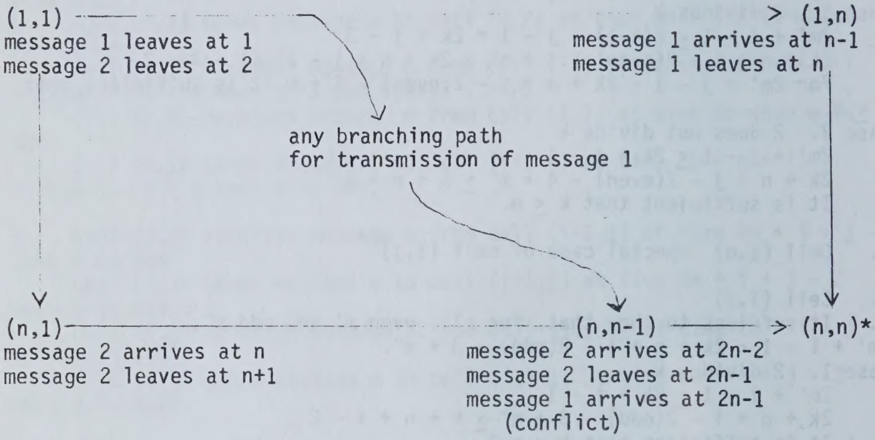
Since no timing conflicts occur when $k < n-1$ for even k or when $k < n$ for odd k, the algorithm succeeds when $k \leq n-2$ for odd n and when $k \leq n-1$ for even n.

A Conjecture Concerning $T(n,k)$

The algorithm described above solves the problem of broadcasting k messages in a grid graph of size n in time $2n + 2k - 4$ for $k \leq n-2$ when n is odd and $k < n-1$ when n is even.

Obviously, $T(n,k) \leq 2n + 2k - 4$ under these conditions. It is conjectured that $T(n,k) = 2n + 2k - 4$, for k "small enough" relative to n.

It can be shown that $T(n,2) > 2n - 1$. Figure 4 illustrates the argument.


(1,1)
message 1 leaves at 1
message 2 leaves at 2
(1,n)
message 1 arrives at n-1
message 1 leaves at n
any branching path
for transmission of message 1
(n,1)
message 2 arrives at n
message 2 leaves at n+1
(n,n-1)
message 2 arrives at 2n-2
message 2 leaves at 2n-1
message 1 arrives at 2n-1
(conflict)
(n,n)*

*at point (n,n)
message 1 arrives at $2n - 2$
message 2 arrives at $2n - 1$

Figure 4
Sketch of Proof That $T(n,2) > 2n - 1$

Wolog assume message 1 leaves point (1,1) at time 1 and travels along row 1 and column n to point (n,n); message 2 leaves point (1,1) at time 2 and travels along column 1 and row n to point (n,n). (This scheme provides for the least interference between messages.) If $T(n,2) \leq 2n-1$, message 1 must reach (n,n) at time $2n - 2$, and message 2 must reach (n,n) at time $2n - 1$. To reach point (n,n-1), message 1 must leave (1,1) after time 2 (too slow) or must follow a path branching from the path through row 1 and column n. Branching from a data path introduces a delay of 1, so message 1 cannot reach point (n,n-1) until $2n - 1$. But point (n,n-1) is busy at time $2n - 1$ transmitting message 2 to point (n,n). So $T(n,2) = 2n$.

REFERENCES

1. P. Chinn, S. Hedetniemi, and S. Mitchell
Multiple message broadcasting.
Proceedings of Tenth Southeastern Conference on Combinatorics, Graph Theory and Computing
Utilitas Mathematica
1979, to appear

2. E. Cockayne and S. Hedetniemi
A conjecture concerning broadcasting in m-dimensional grid graphs.
CS-TR-78-14 Computer Science Department, University of Oregon

3. E. J. Cockayne and A. G. Thomason
Optimal multi-message broadcasting in complete graphs.
DM-169-IR University of Victoria Technical Report

4. A. Farley
Minimal broadcast networks.
Networks
to appear

5. A. Farley
Broadcast time in communication networks.
SIAM Journal of Applied Mathematics
to appear

6. A. Farley
Minimal time line broadcast networks.
Networks
to appear

7. A. M. Farley and S. T. Hedetniemi
Broadcasting in grid graphs.
Proceedings of Ninth Southeastern Conference on Combinatorics,
Graph Theory and Computing
Utilitas Mathematica
1978, pp. 275 - 288

8. A. Farley, S. Hedetniemi, S. Mitchell, and A. Proskurowski
Minimum broadcast graphs.
CS-TR-77-2 Computer Science Department, University of Oregon

9. S. T. Hedetniemi and S. Mitchell
A census of minimum broadcast graphs.
Journal of Combinatorics, Information and Systems Sciences
to appear

10. G. W. Peck
Optimal spreading in n-dimensional rectilinear grids.
submitted

11. A. Proskurowski
Minimum broadcast trees.
CS-TR-78-18 Computer Science Department, University of Oregon

12. P. Slater, E. Cockayne, and S. Hedetniemi
Information dissemination in trees.
CS-TR-78-11 Computer Science Department, University of Oregon

ACKNOWLEDGEMENT

The author wishes to thank Prof. S. T. Hedetniemi of the University of Oregon for suggesting the problem and for providing a list of references.

STRUCTURAL DECOMPOSITION AND STABILIZATION OF LARGE-SCALE CONTROL SYSTEMS*

M. E. Sezer** and D. D. Šiljak
School of Engineering
University of Santa Clara
Santa Clara, California 95053

ABSTRACT

Graph-theoretic concepts are used to decompose a large control system into a block-triangular interconnection of input reachable subsystems. Stabilization of the system is thereby reduced to stabilization of a number of smaller subsystems. The special structural properties of the obtained subsystems can be exploited to further simplify the stabilization problem.

1. INTRODUCTION

Decompositions of dynamic systems are performed either for conceptual or computational reasons. When a system is partitioned into interconnected subsystems, which represent physical entities, the decomposition can expose the important structural features of the overall system. Qualitative aspect of large systems can be considered in this context to determine their performance under planned or accidental changes of the system interconnection structure. Alternatively, a large system can be decomposed into smaller subsystems in order to reduce the dimensionality of the design problems, and provide considerable computational simplifications in a quantitative system analysis. The two aspects of decompositions are not entirely separated, because the computational simplifications can make transparent important conceptual questions raised about the behavior of the overall system.

Research in the past several years [1-12] has shown that the theory of directed graphs and interconnection matrices provides a powerful unifying framework for decompositions of large dynamic systems superceding successfully a whole host of *ad hoc* decomposition schemes and techniques. This is a pleasing development and not completely unexpected [1]. The basic trend in computational simplifications has been to determine stability of a large system by identifying the strongly connected components of the associated directed graph [2-4, 8, 10]. By transforming the system into block-triangular structure where the blocks correspond to the strong components, we conclude stability of the overall system from stability of the subsystems and, possibly, some other easily tested conditions. Aside from certain specific problems, most of the present large-scale systems theory is concerned with problems in control and estimation [9, 13]. In this context, very little is known about the role of the control system structure, and it is unclear how much of the available results about structure of free systems can be used in designing feedback control and estimation schemes. This problem we shall consider in this paper.

---

[*] The research reported herein was supported by the U.S. Department of Energy, Electric Energy Division, under the Contract EC-77-S-03-1493.

[**] On leave from the Electrical Engineering Department, Middle East Technical University, Ankara, Turkey.

The major objective of this work is to propose a graph-theoretic procedure for decomposition of a large control system into a causal (hierarchial) interconnection of input reachable subsystems. Input reachability [5], being the essential requirement for controllability, is a suitable conceptual and computational framework for decomposition of large systems via graph-theoretic techniques for purposes of stabilization. Once a causal structure is obtained, stabilization of the overall system is reduced to stabilization of the subsystems. Input reachability of the subsystems along with another structural condition that can be tested on the subsystem level, is sufficient to ensure stabilizability of the subsystems in all but pathological cases. An effective procedure is also outlined for computing stabilizing feedback control for the subsystems.

Due to space limitations we omit the proofs of all propositions except Lemma 3.6, which is essential in the development of the decomposition scheme. These proofs are available in reference [14].

2. DIRECTED GRAPHS AND CONTROL SYSTEMS

Let us recall several standard definitions in graph theory (e.g. Harary [15]). A *directed graph (digraph)* is a pair $\mathcal{D} = (V, E)$, where V is a set of vertices and E is a set of edges (v_j, v_i) directed from the vertex v_j to the vertex v_i. If $(v_j, v_i) \in E$ then $v_j (v_i)$ is said to be adjacent to v_i (adjacent from v_j). Adjacency relation defines a square binary matrix $M = (m_{ij})$, where $m_{ij} = 1$ if and only if $(v_j, v_i) \in E$, which is called the adjacency or *interconnection matrix.*

A sequence of edges $\{(v_1, v_2), (v_2, v_3), \dots, (v_{k-1}, v_k)\}$ where all the vertices are distinct, is called a path from v_1 to v_k. When v_k coincides with v_1 then the path is called a cycle. A digraph $\mathcal{D}$ is said to be *acyclic* if it contains no cycles. If there is a path from v_j to v_i then we say that v_i is *reachable* from v_j. Similarly, a subset $V_i \subset V$ is reachable from a subset $V_j \subset V$ if every vertex in V_i is reachable from some vertex in V_j. Like adjacency, the reachability relation on V can be represented by a binary matrix $R = (r_{ij})$ such that $r_{ij} = 1$ if and only if v_i is reachable from v_j.

We define a *subgraph* $\mathcal{D}_k = (V_k, E_k)$ of $\mathcal{D}$ as a digraph where $V_k \subset V$ and $E_k \subset E$ such that for any $v_i, v_j \in V_k$ $(v_j, v_i) \in E_k$ if and only if $(v_j, v_i) \in E$. A maximal subgraph of $\mathcal{D}$ in which any two vertices are mutually reachable is called a strong component of $\mathcal{D}$.

We define a *partition* P of a digraph $\mathcal{D}$ as a set of subgraphs $\mathcal{D}_k = (V_k, E_k)$, $k = 1, 2, \dots, s$, such that $V_i \cap V_j = \phi$, $i \neq j$, and $V = \bigcup_{k=1}^{s} V_k$. The *condensation* $\mathcal{D}^*$ of $\mathcal{D}$ with respect to P is a digraph $\mathcal{D}^* = (V^*, E^*)$, where $V^* = \{v_1^*, v_2^*, \dots, v_s^*\}$ and $(v_j^*, v_i^*) \in E^*$ if and only if $v_j^* \neq v_i^*$ and some subset of V_i is reachable from V_j. It is important to note that our definitions of partition and condensation are not standard in the sense that the subgraphs $\mathcal{D}_k$ are not required to be the strong components of $\mathcal{D}$. As we will see later, this modification is essential in our structural analysis of control systems via directed graphs.

Let us consider a control system S described as

$$S\colon \dot{x} = Ax + Bu\,, \tag{2.1}$$

where $x(t) \in R^n$ and $u(t) \in R^m$ are respectively the state and the input of S. With the control system S we associate a directed graph $\mathcal{D} = (U \cup X, E)$ such that $U = \{u_1, u_2, \dots, u_m\}$ and $X = \{x_1, x_2, \dots, x_n\}$ are the sets of input and state vertices, and $(u_j, x_i) \in E$ $[(x_j, x_i) \in E]$ if and only if the input u_j (the state x_j) occurs in the equation corresponding to $\dot{x}_i$ in (2.1). Accordingly, the adjacency and reachability matrices of $\mathcal{D}$ have the form

$$M = \begin{bmatrix} E & L \\ 0 & 0 \end{bmatrix}, \quad R = \begin{bmatrix} P & Q \\ 0 & 0 \end{bmatrix}, \tag{2.2}$$

where E and L (P and Q) are state and input interconnection (reachability) matrices of proper dimensions.

Once the association of a directed graph $\mathcal{D}$ with a given system S is established, the digraph $\mathcal{D}$ can be used to study structural properties of S. In the next section, using graph-theoretic concepts we will exploit this association to decompose the system S into smaller subsystems with such structural properties that allow reduction of dimensionality of the problem of stabilizing the overall system S.

3. INPUT REACHABLE DECOMPOSITIONS

We consider the problem of decomposing the system S of (2.1) into a number of hierarchically ordered subsystems described as

$$S_k\colon \dot{x}_k^* = \sum_{j=1}^{k} A_{kj} x_j^* + \sum_{j=1}^{k} B_{kj} u_j^*\,, \quad k = 1, 2, \dots, s \tag{3.1}$$

where $x_k^*(t) \in R^{n_k}$ and $u_k^*(t) \in R^{m_k}$ are respectively the state and input of S_k, such that $R^n = R^{n_1} \times R^{n_2} \times \dots \times R^{n_s}$, $R^m = R^{m_1} \times R^{m_2} \times \dots \times R^{m_s}$, and A_{kj}, B_{kj} are matrices of proper dimensions. The decomposition (3.1) corresponds to the matrices A and B of (2.1) having the block-triangular form

$$A = \begin{bmatrix} A_{11} & & \bigcirc \\ A_{21} & A_{22} & \\ \vdots & \vdots & \ddots \\ A_{s1} & A_{s2} \cdots & A_{ss} \end{bmatrix}, \quad B = \begin{bmatrix} B_{11} & & \bigcirc \\ B_{21} & B_{22} & \\ \vdots & \vdots & \ddots \\ B_{s1} & B_{s2} \cdots & B_{ss} \end{bmatrix} \tag{3.2}$$

The corresponding interconnection matrix M of (2.2) has the obvious representation

$$M = \left[\begin{array}{cccc|cccc} E_{11} & & \bigcirc & & L_{11} & & \bigcirc & \\ E_{21} & E_{22} & & & L_{21} & L_{22} & & \\ \vdots & \vdots & \ddots & & \vdots & \vdots & \ddots & \\ E_{s1} & E_{s2} & \dots & E_{ss} & L_{s1} & L_{s2} & \dots & L_{ss} \\ \hline & \bigcirc & & & & \bigcirc & & \\ \end{array}\right] \qquad (3.3)$$

In this way, the digraph $\mathcal{D} = (U \cup X, E)$ is partitioned into subgraphs $\mathcal{D}_k = (U_k \cup X_k, E_k)$ identified by the pairs (E_{kk}, L_{kk}) of (3.3). These subgraphs are the digraphs of the decoupled subsystems

$$S_k^o: \dot{x}_k^* = A_{kk}x_k^* + B_{kk}u_k^*, \quad k = 1, 2, \dots, s \qquad (3.4)$$

which are obtained from (3.1) when $A_{kj} = 0$, $B_{kj} = 0$, $j \neq k$.

Due to the block-triangular structure of the matrices A and B in (3.2), the stabilization problem of S can be reduced to stabilization of the decoupled subsystems S_k^o only, provided the subsystems S_k^o are stabilizable. An efficient way to obtain the matrices A and B with such properties is to replace the stabilizability of S_k^o's by input reachability of $\mathcal{D}_k$'s and develop the graph-theoretic techniques to partition a graph $\mathcal{D}$ into input-reachable hierarchically ordered subgraphs $\mathcal{D}_k$. An additional a posteriori analysis which is needed to ensure that input reachability of the subgraphs $\mathcal{D}_k$ imply stabilizability of the subsystems S_k^o, is postponed to Section 4.

We recall [5] that a digraph $\mathcal{D}_k = (U_k \cup X_k, E_k)$ is input reachable if X_k is reachable from U_k, and state the following:

(3.5) DEFINITION. *A partition* P_A *of a digraph* $\mathcal{D} = (U \cup X, E)$ *is said to be an acyclic input reachable partition if:*

(i) *the interconnection matrix* M *has the form* (3.3); *and*

(ii) *each subgraph* $\mathcal{D}_k = (U_k \cup X_k, E_k)$ *is input reachable.*

Of special interest in the following development are *irreducible partitions* P_A which have the property that no subgraphs $\mathcal{D}_k$ can be further partitioned into acyclic input reachable subgraphs. To obtain irreducible partitions P_A we at first assume that the digraph $\mathcal{D}$ is input reachable. Our decomposition scheme, however, can easily be modified to remove the input unreachable part from $\mathcal{D}$ before the decomposition process is started. To avoid trivial cases we assume that each input vertex of $\mathcal{D}$ is adjacent to at least one state vertex. Under these assumptions there always exists an irreducible partition P_A, though that might be the trivial partition with $s = 1$. Obviously, we will be interested in the nontrivial irreducible partitions of $\mathcal{D}$, for the existence of which a necessary condition is that $n > 1$ and $m > 1$. Finally, we note that neither the irreducible partitions nor the number of subgraphs in such partitions need be unique[14].

We now prove a result concerning the existence of acyclic input reachable partitions of a given digraph, which are not necessarily irreducible:

(3.6) LEMMA. *Given a digraph* $\mathcal{D} = (U \cup X, E)$ *with* $n > 1$, $m > 1$. *Then,* $\mathcal{D}$ *has a nontrivial partition* P_A *if and only if the input reachability matrix* Q *in* (2.2) *contains at least one zero element.*

PROOF. Necessity follows by direct computation [7] of Q from M in (3.3). To prove sufficiency, we assume that Q contains a 0 in the (i, j)th position. We partition the set U into two disjoint subsets $U_1 = U - \{u_j\}$ and $U_2 = \{u_j\}$. Let X_2 be the largest subset of X which is reachable from u_j, and let $X_1 = X - X_2$. These subsets have the following properties:

(i) $U_2 \neq \phi$ by construction, and $U_1 \neq \phi$ by the assumption that $m > 1$;

(ii) $X_1 \neq \phi$ for $x_i \in X_1$, and $X_2 \neq \phi$ by the assumption that each input vertex is adjacent to at least one state vertex;

(iii) By construction no $x_\ell \in X_1$ is adjacent from u_j; and

(iv) No $x_\ell \in X_1$ is adjacent from some $x_k \in X_2$, for otherwise x_ℓ would be reachable from u_j contradicting the construction of X_2.

Properties (i)-(iv) imply that corresponding to the chosen subsets of U and X, the interconnection M has the form (3.3) with $s = 2$. Furthermore, both subgraphs $\mathcal{D}_1 = (U_1 \cup X_1, E_1)$ and $\mathcal{D}_2 = (U_2 \cup X_2, E_2)$ are input reachable; $\mathcal{D}_1$ because of the properties (iii) and (iv), and $\mathcal{D}_2$ by construction. Therefore, $\mathcal{D}_1$ and $\mathcal{D}_2$ constitute a partition P_A of the digraph $\mathcal{D}$, and the proof of Lemma (3.6) is complete.

The construction of the partition P_A in the proof of Lemma (3.6) is the core of our decomposition scheme for obtaining the irreducible partitions. This scheme we consider next.

Suppose again $n > 1$, and Q contains at least one 0, as in Lemma (3.6). Let us choose the index sets $I = \{i_1, i_2, \ldots, i_k\} \subset \{1, 2, \ldots, n\}$ and $J = \{j_1, j_2, \ldots, j_\ell\} \subset \{1, 2, \ldots, m\}$ subject to the following conditions.

(i) Rows $i_1, i_2, \ldots, i_k$ of Q are the same, and contain 1 in columns $j_1, j_2, \ldots, j_\ell$ of Q and 0 in other columns;

(ii) when rows $i_1, i_2, \ldots, i_k$ and columns $j_1, j_2, \ldots, j_\ell$ of Q are deleted, the resulting submatrix contains no zero rows.

Note that the assumptions on n, m and Q guarantee the existence of I and J. Now, letting

$$U_1 = \{u_{j_1}, u_{j_2}, \ldots, u_{j_\ell}\}, \quad U_2 = U - U_1, \tag{3.7a}$$

$$X_1 = \{x_{i_1}, x_{i_2}, \ldots, x_{i_k}\}, \quad X_2 = X - X_1, \tag{3.7b}$$

we state the following result:

(3.8) THEOREM. *The subgraphs* $\mathcal{D}_k = (U_k \cup X_k, E_k)$, $k = 1, 2$, *where* U_k *and* X_k *are defined in* (3.7), *constitute a partition* P_A *of* $\mathcal{D} = (U \cup X, E)$ *such that* $\mathcal{D}_1$ *is irreducible.*

The decomposition scheme leading to Theorem (3.8), if applied recursively in such a way that at each step an irreducible subgraph is removed from $\mathcal{D}$, the existence of which is guaranteed by Theorem (3.8), produces an irreducible partition P_A of $\mathcal{D}$. Moreover, if at each step we consider all possible choices of subgraphs to be removed, we can generate all irreducible partitions.

The computations involved in the above decomposition scheme can be considerably reduced by considering a condensation of $\mathcal{D}$ instead of $\mathcal{D}$ itself. For this purpose, we partition the set X and U of $\mathcal{D}$ as

$$X = \bigcup_{i=1}^{p} \bar{X}_i \,, \quad U = \bigcup_{i=1}^{q} \bar{U}_i \tag{3.9}$$

such that $\bar{X}_i$'s are vertex sets of the strong components of the subgraph $\mathcal{D}_x = (X, E_x)$, which was termed the state truncation of $\mathcal{D}$ [7]. Each $\bar{U}_i$ in (3.9) contains all the input vertices which are adjacent exactly to the same vertices, that is, there is no state vertex which is adjacent from an input vertex in $\bar{U}_i$ but is not adjacent from all the vertices in $\bar{U}_i$. We denote by $\mathcal{D}^* = (U^* \cup X^*, E^*)$ the condensation with respect to the partitions of the sets X and U in (3.9), and state the following:

(3.10) THEOREM. *To any irreducible partition* P_A^* *of* $\mathcal{D}^*$ *into* s *subgraphs, there corresponds a unique partition* P_A *of* $\mathcal{D}$ *into the same number of subgraphs, and vice versa.*

Theorem (3.10) simply states that the decomposition problems for $\mathcal{D}$ and $\mathcal{D}^*$ are interchangable. However, since $\mathcal{D}^*$ is considerably smaller than $\mathcal{D}$, irreducible partitions P_A of $\mathcal{D}$ can be generated much more easily by first obtaining $\mathcal{D}^*$ and then applying the decomposition scheme to $\mathcal{D}^*$. An additional advantage of following such a procedure is that the strong components of the subgraphs $\mathcal{D}_k$ would be identified during the decomposition process, allowing the stabilization scheme considered in the next section to be applied to the subsystems S_k^o directly.

4. SUBSYSTEM STABILIZATION

The proposed decomposition scheme was developed with the objective of stabilizing a system by stabilizing a number of smaller input reachable subsystems which are arranged in a hierarchical structure. To be able to stabilize the subsystems with prescribed degree of stability, we need controllability of each subsystem. In the context of our decomposition scheme, it is natural to replace controllability by its structural counterpart - structural controllability [16]. A possible loss of controllability of a structurally controllable subsystem can occur only in pathological cases when there is an exact matching of subsystem parameters. We further note that once the input reachability of the subsystems is established by the decomposition scheme, the structural controllability of the subsystems follows provided an additional structural condition is satisfied: The subsystem digraph $\mathcal{D}_k$ has no dilation [6, 16], or equivalently, the $n_k \times (n_k + m_k)$ matrix $[E_{kk} \; L_{kk}]$ does not contain a zero submatrix of order

$r \times (n_k + m_k + 1 - r)$ or larger for no r such that $1 \leq r \leq n_k$, where E_{kk}, L_{kk} are defined in (3.3). Thus, if no subgraph of an irreducible partition contains a dilation, which can easily be checked using standard algorithms [14, 17], then we can proceed directly to the stabilization procedure. If, on the other hand, some subgraphs turn out to have dilations, then provided that the overall system is structurally controllable, we can modify the partition by combining certain subgraphs suitably in such a way that we avoid the dilations appearing in the subgraphs of the modified partition [14].

We now concentrate on the problem of stabilizing the decoupled subsystems S_k^o in (3.4), which we assume, in the light of the above discussion, to be controllable. For this purpose, let us describe a typical subsystem of (3.4) as

$$S^o: \dot{x}^* = Ax^* + Bu^* , \tag{4.1}$$

by dropping the subscripts appearing in (3.4) for convenience. Controllability of S^o implies that there exists a feedback law

$$u^* = Kx^* , \tag{4.2}$$

which places the eigenvalues of S^o at prescribed locations. Recalling that in the process of obtaining the acyclic decomposition of the overall system, the strong components of each subsystem have also been identified, so that the A matrix in (4.1) is already in block triangular form, we can give an efficient procedure to obtain the feedback matrix K in (4.2).

To begin with, let us first assume that the subsystem S^o in (4.1) has two strong components, so that the A matrix is of the form

$$A = \begin{bmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{bmatrix} . \tag{4.3}$$

We partition the matrices B and K in (4.1) and (4.2) accordingly as

$$B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} , \quad K = [K_1 \quad K_2] . \tag{4.4}$$

From (4.3), it follows that controllability of the pair (A, B) implies controllability of the pair (A_{11}, B_1). Therefore, there exists a matrix F_1 such that

$$\lambda(A_{11} + B_1F_1) = \mathcal{L}_1 , \tag{4.5}$$

where $\lambda(A_{11} + B_1F_1)$ denotes the set of eigenvalues of $A_{11} + B_1F_1$, and $\mathcal{L}_1$ is a symmetric set of prescribed eigenvalues for $A_{11} + B_1F_1$ such that

$$\lambda(A_{22}) \cap \mathcal{L}_1 = \phi . \tag{4.6}$$

We let $A_{11}^c = A_{11} + B_1F_1$, and consider the equation

$$A_{22}T_1 - T_1A_{11}^c = -A_{21} - B_2F_1 . \tag{4.7}$$

Equations (4.5) and (4.6) guarantee that (4.7) has a unique solution for

T_1 [18]. We now state the following:

(4.8) LEMMA. *The pair* $(A_{22}, B_2 - T_1B_1)$ *is controllable.*

Lemma (4.8) implies that, for any symmetric set L_2 of prescribed eigenvalues there exists K_2 such that

$$\lambda[A_{22} + (B_2 - T_1B_1)K_2] = L_2 \ . \tag{4.9}$$

With K_2 chosen as above, we set

$$K_1 = F_1 - K_2T_1 \ , \tag{4.10}$$

and state the following theorem.

(4.11) THEOREM. $(A + BK) = L_1 \cup L_2$, *where* A, B, K, L_1 *and* L_2 *are defined by* (4.3)-(4.5), (4.9) *and* (4.10).

The argument leading to Theorem (4.11) provides a recursive procedure for assigning the eigenvalues of a subsystem having two strong components. Since this procedure requires only transforming (A_{11}, B_1) and $(A_{22}, B_2 - T_1B_1)$ into suitable forms to satisfy (4.5) and (4.9), and solving the equation (5.7), it is more efficient than any single step procedure which requires transforming the whole subsystem. To complete our discussion we now consider the general case, where the matrices A and B of (4.1) have the form

$$A = \left[\begin{array}{c:cccc} A_{11} & 0 & \cdots & & 0 \\ \hdashline A_{21} & A_{22} & & \bigcirc & \\ \vdots & \vdots & \ddots & & \\ A_{k1} & A_{k2} & \cdots & & A_{kk} \end{array}\right] = \begin{bmatrix} \hat{A}_{11} & 0 \\ \hat{A}_{21} & \hat{A}_{22} \end{bmatrix} \tag{4.12}$$

$$B = \left[\begin{array}{c} B_1 \\ \hdashline B_2 \\ \vdots \\ B_k \end{array}\right] = \begin{bmatrix} \hat{B}_1 \\ \hat{B}_2 \end{bmatrix} \tag{4.13}$$

Partitioning the feedback matrix K accordingly as

$$K = [K_1 \,\vdots\, K_2 \cdots K_k] = [\hat{K}_1 \ \ \hat{K}_2] \ , \tag{4.14}$$

and noting that the A matrix in (4.12) has the form (4.3), we observe that eigenvalue assignment problem can be solved following the steps below:

(i) Choose $\hat{F}_1$ such that $\lambda(\hat{A}_{11} + \hat{B}_1\hat{F}_1) = \hat{L}_1$, where $\hat{L}_1$ is a symmetric set of eigenvalues such that $\lambda(\hat{A}_{22}) \cap \hat{L}_1 = \phi$.

(ii) Solve

$$\hat{A}_{22}\hat{T}_1 - \hat{T}_1\hat{A}^c_{11} = -\hat{A}_{21} - \hat{B}_2\hat{F}_1 \tag{4.15}$$

for $\hat{T}_1$, where $\hat{A}^c_{11} = \hat{A}_{11} + \hat{B}_1\hat{F}_1$.

(iii) Choose $\hat{K}_2$ such that $\lambda[\hat{A}_{22} + (\hat{B}_2 - \hat{T}_1\hat{B}_1)\hat{K}_2] = \hat{L}_2$, where $\hat{L}_2$ is a symmetric set of eigenvalues.

(iv) Compute $\hat{K}_1 = \hat{F}_1 - \hat{K}_2\hat{T}_1$. Then, $\lambda(A + BK) = \hat{L}_1 \cup \hat{L}_2$.

Among the above steps only (ii) and (iii) seem to involve operations on relatively large matrices. However, writing (4.15) explicitly as

$$\begin{bmatrix} A_{22} & & \bigcirc \\ \vdots & \ddots & \\ A_{k2} & \cdots & A_{kk} \end{bmatrix} \begin{bmatrix} T_{12} \\ \vdots \\ T_{1k} \end{bmatrix} - \begin{bmatrix} T_{12} \\ \vdots \\ T_{1k} \end{bmatrix} \hat{A}^c_{11} = - \begin{bmatrix} A_{21} + B_2\hat{F}_1 \\ \vdots \\ A_{k1} + B_k\hat{F}_1 \end{bmatrix}, \tag{4.16}$$

we observe that the blocks $T_{12}, \ldots, T_{1k}$ of $\hat{T}_1$ can be obtained by recursive solution of the equations

$$A_{22}T_{12} - T_{12}\hat{A}^c_{11} = -A_{21} - B_2\hat{F}_1,$$
$$A_{ii}T_{1i} - T_{1i}\hat{A}^c_{11} = -A_{i1} - B_i\hat{F}_1 - \sum_{j=2}^{i-1} A_{ij}T_{ij}, \quad i = 3, 4, \ldots, k. \tag{4.17}$$

Also, the problem of computing $\hat{K}_2$ in Step (iii) is similar to the original problem of computing K where A, B and K are replaced by $\hat{A}_{22}$, $\hat{B}_2 - \hat{T}_1\hat{B}_1$ and $\hat{K}_2$ respectively. Thus, the blocks $K_1, K_2, \ldots, K_k$ of the required feedback matrix can be obtained recursively by solving a set of recursive equations as in (4.17) at each step.

5. CONCLUSIONS

A decomposition of a large control system into a block triangular structure of input reachable subsystems can be useful in several ways. First, the stabilization problem of the overall system is reduced to stabilization of a number of smaller problems associated with the subsystems. Second, the special structure of the subsystems allows for a "piece-by-piece" computation of the feedback gains providing an additional possibility for dimensionality reduction of the stabilization problem. Furthermore, once the block triangular structure is determined, the interconnections which are outside of the subsystems can be identified as interconnections which have no effect on stability of the overall closed-loop system. It should also be mentioned that the decomposition scheme being independent of the nature of the equations (nonlinear, discrete, etc.), can be used for stabilization of wider classes of systems than presently considered, provided the obtained subsystems can be stabilized by known methods.

REFERENCES

1. Šiljak, D. D., "Stability of Large-Scale Systems", Proceedings of the Fifth IFAC Congress, Paris, 1972, pp. C-32: 1-11.

2. Kevorkian, A. K., "Structural Aspects of Large Dynamic Systems", Proceedings of the Sixth IFAC Congress, Boston, Massachusetts, 1975, pp. 19:3:1-7.

3. Özgüner, U., and W. R. Perkins, "Graph Theory in the Analysis of Large Scale Composite Dynamic Systems", Proceedings of the International Symposium on Circuits and Systems, Boston, Massachusetts, 1975, pp. 121-123.

4. Mayeda, W., and N. Wax, "System Structure and Stability", Proceedings of the Tenth Annual Asilomar Conference on Circuits, Systems, and Computers, Pacific Grove, California, 1976, pp. 181-185.

5. Šiljak, D. D., "On Reachability of Dynamic Systems", International Journal of Systems Science, 8(1977), 321-338.

6. Davison, E. J., "Connectability and Structural Controllability of Composite Systems", Automatica, 13(1977), 109-123.

7. Šiljak, D. D., "On Pure Structure of Dynamic Systems", Nonlinear Analysis, Theory, Methods, and Applications, 1(1977), pp. 397-413.

8. Callier, F. M., W. S. Chan, and C. A. Desoer, "Input-Output Stability of Interconnected Systems Using Decompositions: An Improved Formulation", IEEE Transactions, AC-23 (1978), pp. 150-163.

9. Šiljak, D. D., "Large-Scale Dynamic Systems: Stability and Structure", North-Holland, New York, 1978.

10. Michel, A. N., R. K. Miller, and W. Tang, "Lyapunov Stability of Interconnected Systems: Decomposition into Strongly Connected Subsystems", IEEE Transactions, CAS-25(1978), pp. 799-809.

11. Frankensen, O. I., P. Falser, and F. J. Evans, "Qualitative Aspects of Large Scale Systems", Springer-Verlag, New York, 1979.

12. Sezer, M. E., and Ö. Hüseyin, "On Decentralized Stabilization of Interconnected Systems", to appear in Automatica, 2(1980).

13. Sandell, Jr., N. R., P. Varaiya, M. Athans, and M. G. Safonov, "Survey of Decentralized Control Methods for Large Scale Systems", IEEE Transactions, AC-23(1978), pp. 108-128.

14. Sezer, M. E., and D. D. Šiljak, "Structural Decomposition and Stabilization of Large-Scale Control Systems", Internal Report EC-77-S-03-1493, University of Santa Clara, Santa Clara, California, 1979.

15. Harary, F., R. Z. Norman, and D. Cartwright, "Structural Models: An Introduction to the Theory of Directed Graphs", Wiley, New York, 1965.

16. Lin, C-T., "Structural Controllability", IEEE Transactions, AC-19(1974), pp. 201-208.

17. Shields, R. W., and J. B. Pearson, "Structural Controllability of Multi-input Linear Systems", IEEE Transactions, AC-21(1976), pp. 203-212.

18. Gantmacher, F. R., "The Theory of Matrices", vol. 1, Chelsea, New York, 1960.

DECENTRALIZED STABILIZATION OF LARGE SCALE SYSTEMS WITH TIME DELAY

M. IKEDA and D. D. ŠILJAK
School of Engineering, University of Santa Clara
Santa Clara, California

ABSTRACT

A class of linear interconnected systems with time delay is defined, which can always be stabilized by local feedback without delay. Any prescribed degree of stability can be achieved by an appropriate choice of the feedback gains.

1. INTRODUCTION

There have been a number of works concerned with stabilization of large scale systems [1-5], where systems containing no time delay are extensively considered, and classes of interconnections among subsystems are given for which a decentralized stabilizing control always exists. Of much interest to this subject is the case of systems with time delay, since large scale systems often meet delay on account of the time required for information processing and its transfer among subsystems. In this paper, we provide a class of large scale delay systems which can always be stabilized by local feedback. The class is characterized by the interconnection pattern among subsystems, regardless of the delay times or strength of interactions. This result is obtained by observing the fact that if a system contains only loops with gains which can be made as small as desired by the choice of local feedback gains, then the system is decentrally stabilizable.

2. MAIN RESULT

We consider an input-decentralized large scale system S composed of s interconnected subsystems

$$S_i:\ \dot{x}_i(t) = A_i x_i(t) + \sum_{j=1}^{s} A_{ij} x_j(t) + \sum_{j=1}^{s} D_{ij} x_j(t-h_{ij}) + b_i u_i(t)$$

$$i = 1, 2, \ldots, s \qquad (2.1)$$

where $x_i(t)$ is an n_i vector and $u_i(t)$ is the scalar input of the subsystem S_i. In (2.1), h_{ij} is a positive number which denotes the delay time, A_i and b_i are a constant $n_i \times n_i$ matrix and n_i vector given in the companion form

$$A_i = \left[\begin{array}{cccc} 0 & 1 & \ldots & 0 \\ 0 & 0 & \ldots & 0 \\ \hdashline 0 & 0 & \ldots & 1 \\ -a_1^i & -a_2^i & \ldots & -a_{n_i}^i \end{array}\right], \quad b_i = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \qquad (2.2)$$

and $A_{ij} = (a_{pq}^{ij})$, $D_{ij} = (d_{pq}^{ij})$ are constant $n_i \times n_j$ matrices.

In order to decentrally stabilize the overall system S, we apply the local feedback control law

$$u_i(t) = -k_i^T x_i(t), \quad i = 1, 2, \ldots, s \qquad (2.3)$$

and get the closed loop system $\hat{S}$ consisting of interconnected subsystems

$$\hat{S}_i: \dot{x}_i(t) = (A_i - b_i k_i^T)x_i(t) + \sum_{j=1}^{s} A_{ij}x_j(t) + \sum_{j=1}^{s} D_{ij}x_j(t-h_{ij})$$

$$i = 1, 2, \dots, s \qquad (2.4)$$

where k_i are constant n_i vectors.

We use the following:

(2.5) DEFINITION. We say that the system S of (2.1) is decentrally stabilizable if there exists a control law (2.3) such that the closed loop system $\hat{S}$ of (2.4) is exponentially stable.

To determine the class of systems that can be stabilized in the sense of Definition (2.5), we characterize the structure of interconnections using the interconnection matrices $E_{ij} = (e^{ij}_{pq})$ which are induced by the matrices A_{ij} and D_{ij} as

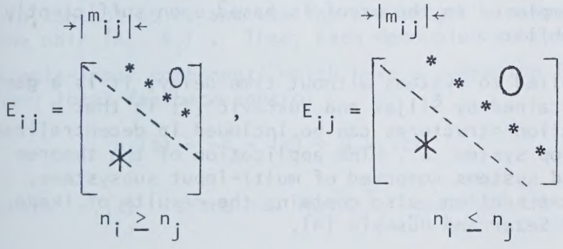
$$e^{ij}_{pq} = \begin{cases} 1, & a^{ij}_{pq} \neq 0 \text{ or } d^{ij}_{pq} \neq 0 \\ 0, & a^{ij}_{pq} = d^{ij}_{pq} = 0 . \end{cases} \qquad (2.6)$$

It should be noted that the matrix E_{ij} represents the pattern of interaction between the subsystems S_i and S_j; the element $e^{ij}_{pq} = 1$ if the q-th element x^j_q of the vector x_j acts on the p-th element x^i_p of the vector x_i, and $e^{ij}_{pq} = 0$ if x^j_q does not act on x^i_p. With each E_{ij} we associate the integer m_{ij} as

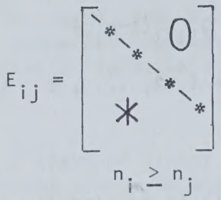
$$m_{ij} = \begin{cases} \max\limits_{(p,q):e^{ij}_{pq} \neq 0} q-p, & E_{ij} \neq 0 \\ -n, & E_{ij} = 0, \end{cases} \qquad (2.7)$$

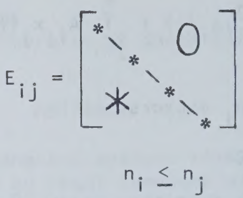
where $n = \sum_{i=1}^{s} n_i$. The integers m_{ij} can be interpreted in terms of the structure of E_{ij} as the distance between the main diagonal and a border line of the nonzero elements in E_{ij}, which is parallel to the main diagonol. The three characteristic cases are as follows:

(i) $m_{ij} > 0$

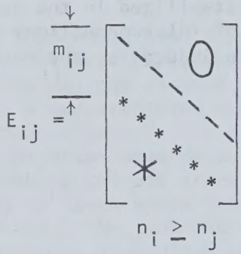


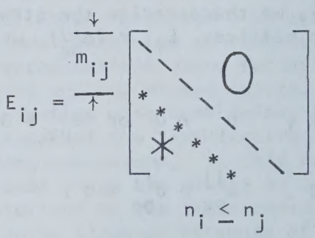
(ii) $m_{ij} = 0$





(iii) $m_{ij} < 0$





(2.8)

In (2.8), the broken line represents the main diagonal of E_{ij}. The border line is indicated by a string of stars each representing the corresponding elements e_{pq}^{ij} of E_{ij} which are not all zero.

We also need to define a subset $I = \{i_1, i_2, \dots, i_r\}$, of the index $\Sigma = \{1, 2, \dots, s\}$, for which $i_1 < i_2 < \dots < i_r$, $r \leq s$. By permuting the elements of I we form a set $J = \{j_1, j_2, \dots, j_r\}$ which is also a subset of Σ but may not be ordered as I.

Our main result is:

(2.9) THEOREM. The system S of (2.1) is decentrally stabilizable if the inequality

$$\sum_{k=1}^{r} (m_{i_k j_k} - 1) < 0 \qquad (2.10)$$

holds for all subsets I and J of the set Σ.

The proof of Theorem (2.9) is outlined in the Appendix. The decentralized stabilization policy employed in the proof is based upon sufficiently high degree of subsystem stability.

When the theorem is applied to systems without time delay, it is a generalization of the result obtained by Šiljak and Vukčević [2] in that a broader class of interconnection structures can be included in decentralized stabilization of the open loop system S. The application of the theorem to the case of interconnected systems composed of multi-input subsystems, which is discussed in the next section, also contains the results of Ikeda, Umefuji, and Kodama [3], and Sezer and Hüseyin [4].

Although the condition of Theorem (2.9) involves $\sum_{r=1}^{s} s!/(s-r)!$ inequalities, the testing can readily be programmed for machine calculations. These computations are not required if $m_{ij} \le n_j - n_i$ for all $i, j = 1, 2, \ldots, s$. This holds whenever the columns of A_{ij} and D_{ij} are proportional to b_i, which means the interactions affect each subsystem through its input.

We should also note that if in (2.1), $A_{ij} = D_{ij} = 0$ for $i \le j$, and we have a "lower-triangular" system [7, 8], then the condition (2.10) of Theorem (2.9) always holds, and system S of (2.1) is automatically stabilizable by decentralized feedback. Furthermore, if groups of subsystems form a lower triangular structure, then our stabilizability condition for the overall system is implied by the stabilizability condition for each group. This fact can be used to reduce considerably the number of inequalities involved in testing the overall stabilizability condition.

To interpret the stabilizability condition (2.10), we associate with the system (2.1) a directed graph as proposed in [6]: the nodes of the graph denote the subsystems, while the edges represent the nonzero interconnections. The condition (2.10) guarantees that the gain of each loop in the graph can be made arbitrarily small by choosing appropriately the gains of the decentralized control law (2.3). Furthermore, this is true regardless of the delay times or magnitude of interconnections. This fact is used to prove Theorem (2.9) in the Appendix.

The overall system can be stabilized to have any degree of exponential stability at the cost of the magnitude of the feedback gains k_i in the control (2.3).

3. MULTI-INPUT CASE

Our main result can be generalized for the case of an interconnected system $\tilde{S}$ composed of multi-input subsystems described by the equations

$$\tilde{S}_i\colon \dot{x}_i(t) = A_i x_i(t) + \sum_{j=1}^{s} A_{ij} x_j(t) + \sum_{j=1}^{s} D_{ij} x_j(t-h_{ij}) + B_i u_i(t)\,, \quad i = 1, 2, \ldots, s \tag{3.1}$$

where $u_i(t)$ is the input ℓ_i vector of the i-th subsystem, and B_i is an $n_i \times \ell_i$ constant matrix. For the purpose of stabilization, we assume that the decoupled subsystems

$$\tilde{S}_i^o\colon \dot{x}_i(t) = A_i x_i(t) + B_i u_i(t)\,, \quad i = 1, 2, \ldots, s \tag{3.2}$$

are controllable, and use the Luenberger's canonical form [9] to represent the pair (A_i, B_i). Then, each decoupled subsystem is reducible to a set of single-input components which have a companion form, by applying a preliminary local feedback control

$$u_i(t) = -\tilde{K}_i x_i(t) + \tilde{G}_i v_i(t) \tag{3.3}$$

where $v_i(t)$ is the new input $\tilde{\ell}_i$ vector of the i-th subsystem, $\tilde{K}_i$ is an

$\ell_i \times n_i$ gain matrix, $\tilde{G}_i$ is an $\ell_i \times \tilde{\ell}_i$ constant matrix, and the integer $\tilde{\ell}_i$ is the number of the single-input components produced by the subsystems $\tilde{S}_i$. That is, the closed loop system $\bar{S}$ with

$$\bar{S}_i: \dot{x}_i(t) = (A_i - B_i\tilde{K}_i)x_i(t) + \sum_{j=1}^{s} A_{ij}x_j(t) + \sum_{j=1}^{s} D_{ij}x_j(t-h_{ij}) + B_i\tilde{G}_i v_i(t), \qquad i = 1, 2, \dots, s \tag{3.4}$$

which corresponds to the preliminary control (3.3), becomes an input decentralized large-scale system each single-input component of which is of the form (2.1). Now we can apply Theorem (2.9) to the system $\bar{S}$ of (3.4). This reduction of multi-input subsystems to single-input components via Luenberger's canonical form, as well as the form itself, are not unique. More details on the reduction are given in [3, 4].

To provide an application of our main result of Theorem (2.9) and, at the same time, illustrate its extension to multi-input subsystems, let us consider a system $\tilde{S}$ composed of two interconnected subsystems

$$\tilde{S}_1: \dot{x}_1(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 2 & 4 & 6 & 8 \\ 0 & 0 & 0 & 1 \\ 3 & 5 & 7 & 9 \end{bmatrix} x_1(t) + \begin{bmatrix} 2 & 3 \\ 4 & 5 \\ 6 & 0 \\ 8 & 9 \end{bmatrix} x_2(t) + \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} u_1(t)$$

$$\begin{aligned} \tilde{S}_2: \dot{x}_2(t) &= \begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix} x_2(t) + \begin{bmatrix} 2 & 0 & 0 & 0 \\ 4 & 5 & 0 & 0 \end{bmatrix} x_1(t) \\ &\quad + \begin{bmatrix} 0 & 0 & 6 & 7 \\ 0 & 0 & 8 & 9 \end{bmatrix} x_1(t-h_{21}) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_2(t) \end{aligned} \tag{3.5}$$

where decoupled subsystems $\tilde{S}_1^o$ and $\tilde{S}_2^o$ are given in the Luenberger's canonical form. To transform $\tilde{S}_1$ into a single-input subsystem, we implement the local control law defined by

$$u_1(t) = -\begin{bmatrix} 2 & 4 & 5 & 8 \\ 0 & 0 & 0 & 0 \end{bmatrix} x_1(t) + \begin{bmatrix} -1 \\ 1 \end{bmatrix} v_1(t) \tag{3.6}$$

and get the subsystem

$$\bar{S}_1: \dot{x}_1(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 3 & 5 & 7 & 9 \end{bmatrix} x_1(t) + \begin{bmatrix} 2 & 3 \\ 4 & 5 \\ 6 & 0 \\ 8 & 9 \end{bmatrix} x_2(t) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_1(t). \tag{3.7}$$

For the system $\bar{S}$ composed of $\bar{S}_1$ and $\bar{S}_2$, which is identical to $\tilde{S}_2$, the interconnection matrices E_{ij} are calculated as

$$E_{11} = 0, \quad E_{22} = 0, \quad E_{12} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}, \quad E_{21} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \tag{3.8}$$

so that the associated integers m_{ij} are

$$m_{11} = -6 \,, \quad m_{22} = -6 \,, \quad m_{12} = 1 \,, \quad m_{21} = 3 \,. \tag{3.9}$$

Since

$$(m_{12} - 1) + (m_{21} - 1) = 2 > 0 \,, \tag{3.10}$$

this system does not satisfy the condition of Theorem (2.9), and we cannot establish decentral stabilizability of the system $\tilde{S}$.

We next consider the local control law defined by

$$u_1(t) = - \begin{bmatrix} -3 & -5 & 6 & 8 \\ 3 & 5 & 0 & 0 \end{bmatrix} x_1(t) + \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} v_1(t) \tag{3.11}$$

to reduce the subsystem $\tilde{S}_1$ to two single-input components

$$\bar{S}_{11}\colon \dot{x}_{11}(t) = \begin{bmatrix} 0 & 1 \\ 2 & 4 \end{bmatrix} x_{11}(t) + \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix} x_2(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v_{11}(t)$$

$$\bar{S}_{12}\colon \dot{x}_{12}(t) = \begin{bmatrix} 0 & 1 \\ 7 & 9 \end{bmatrix} x_{12}(t) + \begin{bmatrix} 6 & 0 \\ 8 & 9 \end{bmatrix} x_2(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v_{12}(t) \tag{3.12}$$

where $x_1 = (x_{11}^T, x_{12}^T)^T$ and $v_1 = (v_{11}, v_{12})^T$. Then, $\tilde{S}_2$ is described by

$$\bar{S}_2\colon \dot{x}_2(t) = \begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix} x_2(t) + \begin{bmatrix} 2 & 0 \\ 4 & 5 \end{bmatrix} x_{11}(t) + \begin{bmatrix} 6 & 7 \\ 8 & 9 \end{bmatrix} x_{12}(t-h_{21}) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_2(t) \,. \tag{3.13}$$

Now we reorder the subsystems $\bar{S}_{11}, \bar{S}_{12}, \bar{S}_2$ as S_1, S_2, S_3 to form the system S, for which the interconnection matrices are

$$E_{13} = E_{32} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} , \quad E_{23} = E_{31} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} , \quad E_{ij} = 0 \text{ otherwise,} \tag{3.14}$$

and the associated integers are

$$m_{13} = m_{32} = 1 \,, \quad m_{23} = m_{31} = 0 \,, \quad m_{ij} = -6 \text{ otherwise} \,. \tag{3.15}$$

Since the integers (3.15) satisfy the condition of Theorem (2.9), we see that the system S is stabilizable by the local feedback

$$v_{11}(t) = -k_{11}^T x_{11}(t) \,, \quad v_{12}(t) = -k_{12}^T x_{12}(t) \,, \quad u_2(t) = -k_2^T x_2(t) \tag{3.16}$$

By going back to the original system $\tilde{S}$, we conclude that $\tilde{S}$ is stabilized by the local feedback

$$\begin{aligned} u_1(t) &= - \left(\begin{bmatrix} -3 & -5 & 6 & 8 \\ 3 & 5 & 0 & 0 \end{bmatrix} + \begin{bmatrix} k_{11}^T & -k_{12}^T \\ 0 & k_{12}^T \end{bmatrix} \right) x_1(t) \\ u_2(t) &= -k_2^T x_2(t) \,. \end{aligned} \tag{3.17}$$

To complete the design, we choose the gains k_{11}, K_{12}, and k_2 by the standard single-input pole-placement scheme.

4. CONCLUDING REMARKS

The class of linear interconnected systems which can always be stabilized by local feedback, has been broadened and generalized to include systems with time delay. This class is characterized by the pure structure of interconnections which does not involve the delay time or strength of interactions. It is straightforward to extend the results to the case of multiple delay times in individual interconnections.

ACKNOWLEDGEMENT

This work reported herein was supported by the U.S. Department of Energy, Electric Energy Division, under Contract EC-77-S-03-1493. The authors express their gratitude to Professor M. E. Sezer, Middle East Technical University, Ankara, Turkey (presently a visiting research associate at the School of Engineering, University of Santa Clara, Santa Clara, California, U.S.A.), for his helpful comments on this work.

REFERENCES

[1] E. J. Davison: The decentralized stabilization and control of a class of unknown nonlinear time-varying systems. *Automatica*, vol. 10, pp. 309-316, May 1974.

[2] D. D. Šiljak and M. B. Vukčević: Decentrally stabilizable linear and bilinear large-scale systems. *International Journal of Control*, vol. 26, pp. 289-305, Aug. 1977.

[3] M. Ikeda, O. Umefuji and S. Kodama: Stabilization of large-scale linear systems. *Transactions IECE of Japan (D)*, vol. 59-D, pp. 355-362, May 1976; also in *Systems, Computers, and Control*, vol. 7, pp. 34-41, 1976.

[4] M. E. Sezer and O. Hüseyin: Stabilization of linear time-invariant interconnected systems using local state feedback. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-8, pp. 751-756, Oct. 1978.

[5] M. Ikeda and D. D. Šiljak: Decentralized stabilization of linear large-scale systems. *Proc. Twelveth Asilomar Conference on Circuits, Systems and Computers*, Pacific Grove, Calif., pp. 390-393, Nov. 1978.

[6] D. D. Šiljak: *Large-Scale Dynamic Systems: Stability and Structure*. North-Holland, New York, 1978.

[7] Ü. Özgüner and W. R. Perkins: Structural properties of large-scale composite systems. *Large-Scale Dynamical Systems*, R. Saeks (ed.), Point Lobos Press, N. Hollywood, California, pp. 5-32, 1975.

[8] F. M. Callier, W. S. Chan, and C. A. Desoer: Input-Output stability of interconnected systems using decompositions: An improved formulation. *IEEE Transactions on Automatic Control*, vol. AC-23, pp. 150-163, April 1978.

[9] D. G. Luenberger: Canonical forms for linear multivariable systems. *IEEE Transactions on Automatic Control*, vol. AC-12, pp. 290-293, June 1967.

[10] M. Araki and B. Kondo: Stability and transient behavior of composite nonlinear systems. *IEEE Transactions on Automatic Control*, vol. AC-17, pp. 537-541, Aug. 1972.

APPENDIX

Proof of Theorem (2.9)

Following the development of Šiljak and Vukčević [2] we choose the gain vectors k_i of (2.3) so that each matrix $A_i - b_i k_i^T$ has a set $L_i = \{\lambda_k^i, k = 1, 2, \ldots, n_i\}$ of distinct real eigenvalues λ_k^i defined by

$$L_i = \{-\alpha\sigma_k^i, \quad k = 1, 2, \ldots, n_i\} \tag{A.1}$$

where $\alpha \geq 1$ and $\sigma_k^i > 0$. The positive number α is to be determined so that the overall system (2.4) is exponentially stable. For this purpose, we apply the nonsingular transformation

$$x_i = T_i \tilde{x}_i \tag{A.2}$$

to the system (2.4) and get

$$\dot{\tilde{x}}_i(t) = \Lambda_i \tilde{x}_i(t) + \sum_{j=1}^{s} \tilde{A}_{ij} \tilde{x}_j(t) + \sum_{j=1}^{s} \tilde{D}_{ij} \tilde{x}_j(t-h_{ij})$$

$$s = 1, 2, \ldots, s \tag{A.3}$$

where

$$\Lambda_i = T_i^{-1}(A_i - b_i k_i^T) T_i, \quad \tilde{A}_{ij} = T_i^{-1} A_{ij} T_j, \quad \tilde{D}_{ij} = T_i^{-1} D_{ij} T_j \tag{A.4}$$

and

$$\Lambda_i = \text{diag}\,\{-\alpha\sigma_1^i, -\alpha\sigma_2^i, \ldots, -\alpha\sigma_{n_i}^i\}\,. \tag{A.5}$$

The transformation matrix T_i can be factorized as

$$T_i = R_i \hat{T}_i\,, \tag{A.6}$$

where

$$R_i = \text{diag}\,\{1, \alpha, \ldots, \alpha^{n_i - 1}\}\,, \tag{A.7}$$

and $\hat{T}_i$ is the Vandermonde matrix induced by the set L_i with $\alpha = 1$. To measure the magnitude of interconnection described by $\tilde{A}_{ij}$ and $\tilde{D}_{ij}$, we use the number

$$\xi_{ij} = ||\hat{T}_i^{-1}||\; ||\hat{T}_j|| \sum_{p=1}^{n_i} \sum_{q=1}^{n_j} (|a_{pq}^{ij}| + |d_{pq}^{ij}|) \tag{A.8}$$

for which we have

$$||\tilde{A}_{ij}|| + ||\tilde{D}_{ij}|| \leq \alpha^{m_{ij}} \xi_{ij}\,. \tag{A.9}$$

We define the $s \times s$ matrix

$$P = \text{diag}\,\{\sigma_m^1, \sigma_m^2, \ldots, \sigma_m^s\}\,, \quad Q = (\alpha^{m_{ij}-1} \xi_{ij}) \tag{A.10}$$

where $\sigma_m^i = \min_k \sigma_k^i$, and use the matrix

$$W = P - Q \, , \tag{A.11}$$

to state the following:

(A.12) LEMMA. The system (2.1) is decentrally stabilizable if the leading principal minors of the matrix W are all positive for some $\alpha \geq 1$.

Proof. Since the system (A.3) has the same stability property as the system (2.4), it suffices to show that (A.3) is exponentially stable. For this purpose, we consider the characteristic equation of (A.3),

$$\det \, [F_D(\lambda) - F_C(\lambda)] = 0 \, , \tag{A.13}$$

where $F_D(\lambda) = \mathrm{diag} \, \{\lambda I_{n_1} - \Lambda_1, \lambda I_{n_2} - \Lambda_2, \dots, \lambda I_{n_s} - \Lambda_s\}$, $F_C(\lambda) = (\tilde{A}_{ij} + \tilde{D}_{ij} e^{-\lambda h_{ij}})$, and I_{n_i} is the identity matrix with dimension n_i . It can be shown that if (A.13) has a solution λ with nonnegative real part, then there exists a nonnegative vector $z \neq 0$ such that

$$Wz \leq 0 \, , \tag{A.14}$$

which contradicts the assumption on W [10]. This fact implies that all the solutions of the characteristic equation (A.13) have negative real parts, and proves Lemma (A.12).

We use Lemma (A.12) to prove Theorem (2.9). The condition of the theorem implies that the r-th leading principal minor of the matrix W can be represented as

$$\sigma_m^1 \sigma_m^2 \cdots \sigma_m^r + c_{r1}\alpha^{-1} + c_{r2}\alpha^{-2} + \dots + c_{rn}\alpha^{-n} \, , \tag{A.15}$$

where $c_{r\ell}$'s are constant numbers. This, in turn, implies that every leading principal minor of W is positive for a sufficiently large positive number α . From Lemma (A.12) follows Theorem (2.9), and the proof is completed.

In order to see the possibility of stabilizing the system (2.1) to have any prescribed degree π of stability, we consider a system described by the equation

$$\dot{\tilde{x}}_i(t) = (\Lambda_i + \pi I_{n_i})\tilde{x}_i(t) + \sum_{j=1}^{s} \tilde{A}_{ij}\tilde{x}_j(t) + \sum_{j=1}^{s} e^{\pi h_{ij}} \tilde{D}_{ij}\tilde{x}_j(t-h_{ij}) \tag{A.16}$$

instead of (A.3). We note that if the system (A.16) is exponentially stable, then the system (A.3) has the degree π of stability. Since a similar argument for the above holds for (A.16) by redefining the number ξ_{ij} and matrix W as

$$\xi_{ij} = ||\hat{T}_i^{-1}|| \; ||\hat{T}_j|| \sum_{p=1}^{n_i} \sum_{q=1}^{n_j} (|a_{pq}^{ij}| + e^{\pi h_{ij}} |d_{pq}^{ij}|)$$

$$W = P - \alpha^{-1}\pi I_s - Q \, , \tag{A.17}$$

we conclude that (A.16) is exponentially stable. This implies that by choosing the gains k_i of the local control law (2.3) appropriately, we can realize any degree of stability in the closed loop system (2.4).

CONSTRUCTION OF LYAPUNOV FUNCTIONS FOR COMPOSITE SYSTEMS BASED ON THE FREQUENCY-DOMAIN CONDITION

MASAMI SAEKI, MITUHIKO ARAKI† and BUNJI KONDO
Department of Electrical Engineering, Kyoto University,
Kyoto, Japan

ABSTRACT

This paper is concerned with stability of such composite systems whose interconnections are functions of the scalar outputs of subsystems. Under the assumption that the frequency-domain condition previously obtained by input output consideration is satisfied, such Lyapunov functions of subsystems are contructed that their weighted sum is a Lyapunov function of the overall system. The results cover the case in which some of the linear parts of subsystems are unstable. A method to estimate the domain of attraction based on those Lyapunov functions is also studied. In order to derive the above results, a new stability theorem of composite systems which employs two sets of functions to reduce negative-definiteness of the derivative of the overall Lyapunov function is proved. The results are useful for studying local stability of large-scale systems.

1. INTRODUCTION

In this paper, we will be concerned with the composite system (refered to as the system CS in the following) which is described by the equations

$$\dot{\underline{x}}_j = F_j\underline{x}_j+\underline{b}_j u_j \quad , \quad y_j = \underline{c}_j^T\underline{x}_j \qquad j=1,---,m \tag{1}$$

$$u_j = -\phi_j(y_j,t)+g_j(y_1,---,y_m,t) \qquad j=1,---,m \tag{2}$$

where the scalor-valued functions $\phi_j(y_j,t)$ and $g_j(y_1,---,y_m,t)$ are assumed to satisfy

$$\xi_j y_j^2 \leqq \phi_j(y_j,t)y_j \leqq \eta_j y_j^2 \quad , \quad \phi_j(0,t) = 0 \tag{3}$$

$$|g_j(y_1,---,y_m,t)| \leqq \sum_{k=1}^{m} \beta_{jk}|y_k| \tag{4}$$

for those values of y_j restricted by

$$-\rho_{1j} \leqq y_j \leqq \rho_{2j} \qquad j=1,---,m \tag{5}$$

In the above, u_j and y_j are scalors; $\underline{x}_j$, $\underline{b}_j$ and $\underline{c}_j$ are n_j-vectors; F_j is an $n_j \times n_j$ matrix; η_j, ρ_{1j} and ρ_{2j} are positive constants; β_{jk} is a nonnegative constant; and ξ_j is a constant less than η_j. Let $\underline{x}$ be the state vector of the system CS given by

$$\underline{x} = (\underline{x}_1^T,---,\underline{x}_m^T)^T$$

Then, from (3) and (4), the origin $\underline{x} = \underline{0}$ is an equilibrium point of the system CS. It is stability of this equilibrium which will be studied in the following.

---

† M. Araki is on leave to Department of Electrical Engineering and Computer Science, University of Santa Clara, Santa Clara, California until October, 1979 supported by the U.S. Department of Energy, Electric Energy Division, under Contract EC-77-S-03-1493.

If (3) and (4) are satisfied for all values of y_j without restriction (5), and if the linear parts (1) do not have unstable poles, we can use the simple frequency-domain condition of [1] in order to assure L_2-stability, and also asymptotic stability in the large under a certain controllability and observability condition. The first purpose of this paper is to prove, under the weaker assumptions stated in the above paragraph, that we can construct a Lyapunov function of the overall system as a weighted sum of the Lyapunov functions of subsystems if the frequency-domain condtion parallel to the above mentioned one is satisfied. The relation between the frequency-domain condition and the existence of the above sort of Lyapunov function was first suggested and partially proved in [2] where the major interest lay in showing the parallelism between the Lyapunov stability analysis and the input-output stability analysis. In the present paper we treat the general case which was not treated in [2]; i.e. the case where ξ_j and η_j are arbitrary numbers satisfying $\xi_j<\eta_j$ and $\eta_j>0$, where the linear parts (1) may have unstable poles, and where (3) and (4) are guaranteed only in the finite region (5). We also propose a method to obtain an estimate of the domain of attraction based on the above Lyapunov functions.

2. MAIN RESULTS

Concerning asymptotic stability of the system CS, we have the next three theorms. The first theorem gives a sufficient condition for asymptotic stability in terms of the frequency responses. The second shows a concrete way of constructing a Lyapunov function and the third indicates a method to obtain an estimate of the domain of attraction.

Theorem 1 (Stability Condition): Assume each $(F_j,\underline{b}_j)$ is a controllable pair and let $f_j(s)$ be the transfer function of the linear part (1)

$$f_j(s) = \underline{c}_j^T(sI-F_j)^{-1}\underline{b}_j \tag{6}$$

Then, the origin of the system CS is asymptotically stable if there exists a positive number α_j for each j such that

$$h_j(s) = \frac{1+(\eta_j+\alpha_j)f_j(s)}{1+(\xi_j-\alpha_j)f_j(s)} \tag{7}$$

is positive real, and if the m×m matrix A-B is an M-matrix where

$$A = \mathrm{diag}(\alpha_j), \quad B = (\beta_{jk}) \qquad j,k=1,\cdots,m \tag{8}$$

Theorem 2 (Construction of Lyapunov Function): Assume the requirements of Theorem 1 are satisfied, and let δ be a sufficiently small positive number. Then, there exists a positive-definite $n_j \times n_j$ matrix P_j which satisfies

$$F_j^TP_j+P_jF_j = -\underline{q}_j\underline{q}_j^T+(\xi_j-\alpha_j+\delta)(\eta_j+\alpha_j-\delta)\underline{c}_j\underline{c}_j^T-\varepsilon_jI \tag{9}$$

$$P_j\underline{b}_j-\frac{1}{2}(\xi_j+\eta_j)\underline{c}_j = \underline{q}_j \tag{10}$$

for a sufficiently small positive number ε_j and for some n_j-vector $\underline{q}_j$, there exists a diagonal matrix $D=\mathrm{diag}(d_j)$ with $d_j>0$ which makes $\tilde{M}D\tilde{M}-\tilde{\Gamma}^T D\tilde{\Gamma}$ positive-semidefinite where

$$\tilde{M} = \mathrm{diag}\left(\frac{\eta_j-\xi_j}{2}+\alpha_j-\delta\right), \quad \tilde{\Gamma} = \mathrm{diag}\left(\frac{\eta_j-\xi_j}{2}\right)+B \qquad j=1,---,m \tag{11}$$

, and the function $v(\underline{x})$ defined by

$$v(\underline{x}) = d_1\underline{x}_1^T P_1\underline{x}_1 + --- + d_m\underline{x}_m^T P_m\underline{x}_m \tag{12}$$

is a Lyapunov function of the system CS.

Theorem 3 (Domain of Attraction): Assume the requirements of Theorem 1 are satisfied and let P_j, D and $v(\underline{x})$ be as given in Theorem 2. Then, the region

$$S = \{\underline{x} \mid v(\underline{x}) \leq \pi_0\} \tag{13}$$

is included in the domain of attraction of the origin of the system CS where

$$\pi_0 = \min_{1\leq j\leq m} d_j\pi_j \tag{14}$$

$$\pi_j = \min_{\underline{c}_j^T\underline{x}_j=-\rho_{1j} \text{ or } \rho_{2j}} \underline{x}_j^T P_j\underline{x}_j \tag{15}$$

Theorems 1 and 2 are stating the same fact from different viewpoints. They are stated seperately here in order to make it clear that stability can be assured only from the frequency responses of the linear parts and from the bounds on the nonlinear functions. Proofs will be given in Sec.4.

Theorems 1-3 indicate a procedure of testing local stability and obtaining an estimate of the domain of attraction. In that procedure we must calculate α_j, P_j and D. As for α_j, the condition that $h_j(s)$ should be positive real is nothing but the well-known circle condition. So, α_j can be easily obtained graphically. To obtain P_j and D, we must first determine δ and ε_j. As for δ, an explicit upper bound is given in the proof. As for ε_j, no simple bound is known. So we can only suggest to start with a seemingly small value. To calculate P_j, substitute (10) into (9). Then, we obtain a Riccati type equation with positive quadratic term. This equation can be solved by the Potter's method [11] and has many positive-definite solutions. (In applying the Potter's method, note that the eigenvectors of the Hamilton matrix corresponding to negative eigenvalue can also give positive-definite solutions in this case.) The problem of choosing the best P_j is open. Calculation of D can be done following the proof of Lemma 2.

3. THREE LEMMAS

To prove the main results, we need three lemmas. The first one gives a new stability condition which is applicable to a general class of composite systems. The second is concerned with a property of M-matrices and was already reported in [3]. The third lemma is concerned with the relation of a frequency-domain condition to existence of a positive-definite solution of the equations with the forms (9) and (10). It is essentially equivalent to Yakubovich lemma. The case $\varepsilon > 0$ of this lemma was attributed to Rekasius and Rowland [4] by Narendra and Taylor [5]. A parallel lemma with $Q = 0$ is also found in [6].

Lemma 1: Consider the system described by

$$\dot{\underline{x}}_j = \tilde{\underline{f}}_j(\underline{x}_j, t) + \tilde{\underline{g}}_j(\underline{x}_1, \text{---}, \underline{x}_m, t) \qquad j = 1, \text{---}, m \tag{16}$$

where

$$\tilde{\underline{f}}_j(\underline{0}, t) = \underline{0}\,, \quad \tilde{\underline{g}}_j(\underline{0}, \text{---}, \underline{0}, t) = \underline{0} \qquad j = 1, \text{---}, m\,. \tag{17}$$

The origin of the system (16) is asymptotically stable if the next three conditions are satisfied.

(a) For each j, there is a decrescent, positive-definite function $v_j(\underline{x}_j, t)$ with continuous partial derivatives such that

$$\dot{v}_j(\underline{x}_j, t)\big|_{(19)} \leqq -\kappa_j[u_{1j}(\underline{x}_j)]^2 + 2\lambda_j u_{1j}(\underline{x}_j)\, u_{2j}(\underline{x}_j) - \mu_j[u_{2j}(\underline{x}_j)]^2$$
$$- u_{0j}(\underline{x}_j) \qquad \underline{x}_j \varepsilon R_j \tag{18}$$

where $\dot{v}_j(\underline{x}_j, t)\big|_{(19)}$ is the derivative along the solution of the system

$$\dot{\underline{x}}_j = \tilde{\underline{f}}_j(\underline{x}_j, t)\,, \tag{19}$$

κ_j and μ_j are positive constants, λ_j a nonnegative constant, $u_{1j}(\underline{x}_j)$ and $u_{2j}(\underline{x}_j)$ positive-semidefinite functions, $u_{0j}(\underline{x}_j)$ a positive-definite function and R_j is a set including the origin $\underline{x}_j = \underline{0}$ as an inner point.

(b) There are nonnegative constants γ_{jk} such that

$$[\text{grad}_j v_j(\underline{x}_j, t)]^T \tilde{\underline{g}}_j(\underline{x}_1, \text{---}, \underline{x}_m, t) \leqq 2u_{1j}(\underline{x}_j) \sum_{k=1}^{m} \gamma_{jk} u_{2k}(\underline{x}_k)$$
$$\underline{x}_k \varepsilon R_k \quad \text{for } k = 1, \text{---}, m \tag{20}$$

where grad_j means the gradient with respect to $\underline{x}_j$.

(c) The matrix $K^{\frac{1}{2}}M^{\frac{1}{2}} - (\Lambda + \Gamma)$ is an M-matrix where
$K = \text{diag}(\kappa_j)$, $\Lambda = \text{diag}(\lambda_j)$, $M = \text{diag}(\mu_j)$, $\Gamma = (\gamma_{jk})$ $j, k=1, \text{---}, m$.

[Note] A function $v_j(\underline{x}_j, t)$ is said to be positive-definite if there is a nondecreasing function $v_j'(\sigma)$ of a real variable σ such that $v_j'(0) = 0$, $v_j'(\sigma) > 0$ for $\sigma \neq 0$ and $v_j(\underline{x}_j, t) \geqq v_j'(||\underline{x}_j||)$. It is said to be decrescent if there is a nondecreasing function $v_j''(\sigma)$ such that $v_j''(0) = 0$ and $|v_j(\underline{x}_j, t)| \leq v_j''(||\underline{x}_j||)$.

[Proof] Let $d_j > 0$ and define $v(\underline{x}, t)$ by

$$v(\underline{x}, t) = d_1 v_1(\underline{x}_1, t) + \text{---} + d_m v_m(\underline{x}_m, t)\,. \tag{21}$$

Then, the derivative of v along the solution of the system (16) is given by

$$\dot{v}|_{(16)} = \sum_{j=1}^{m} d_j\{\dot{v}_j|_{(19)} + (\mathrm{grad}_j v_j)^T \tilde{\underline{g}}_j\} \ . \tag{22}$$

By (18) and (20) we obtain

$$\dot{v}|_{(16)} \leqq - \sum_{j=1}^{m} d_j \kappa_j u_{1j}{}^2 + 2 \sum_{j=1}^{m} d_j \lambda_j u_{1j} u_{2j} - \sum_{j=1}^{m} d_j \mu_j u_{2j}{}^2 +$$

$$2 \sum_{j=1}^{m} \sum_{k=1}^{m} d_j \gamma_{jk} u_{1j} u_{2k} - \sum_{j=1}^{m} d_j u_{0j} \qquad \underline{x} \varepsilon R \tag{23}$$

where R is the direct product of R_1, ---, R_m. Here, let $\underline{u}$ be the 2m-vector given by

$$\underline{u} = (u_{11}, \text{---}, u_{1m}, u_{21}, \text{---}, u_{2m})^T \ .$$

Then, eq. (23) can be written as

$$\dot{v}|_{(25)} \leqq -\underline{u}^T V \underline{u} - \sum_{j=1}^{m} d_j u_{0j} \qquad \underline{x} \varepsilon R \tag{24}$$

where V is the $(2m)\times(2m)$ matrix given by

$$V = \begin{pmatrix} DK & -D(\Lambda+\Gamma) \\ -(\Lambda+\Gamma)^T D & DM \end{pmatrix}, \quad D = \mathrm{diag}(d_j) \ ; \ j = 1, \text{---}, m. \tag{25}$$

The above V can be expressed as

$$V = \begin{pmatrix} 1 & 0 \\ (\Lambda+\Gamma)^T K^{-1} & 1 \end{pmatrix}^{-1} \begin{pmatrix} DK & 0 \\ 0 & V_0 \end{pmatrix} \begin{pmatrix} 1 & K^{-1}(\Lambda+\Gamma) \\ 0 & 1 \end{pmatrix}^{-1} \tag{26}$$

where

$$V_0 = M^{\frac{1}{2}} D M^{\frac{1}{2}} - (\Lambda+\Gamma)^T K^{-\frac{1}{2}} D K^{-\frac{1}{2}} (\Lambda+\Gamma) \ . \tag{27}$$

By assumption (c) and by the property of M-matrices (Th. 5 of [7]), the matrix $M^{\frac{1}{2}}-K^{-\frac{1}{2}}(\Lambda+\Gamma)$ is an M-matrix. Therefore, by Lemma 2 below, we can choose D so that V_0 is positive-definite. Then, the positive-definiteness of V follows from (26). Therefore, the righthand side of (24) is negative-definite for $\underline{x}\varepsilon R$ (note that the second term is negative-definite whereas the first term is negative-semidefinite). Since $v(\underline{x}, t)$ is positive-definite and decrescent from the definition (21), and since R is a set including the origin $\underline{x} = \underline{0}$ as an inner point, the origin of the system (16) is asymptotically stable. [Q.E.D]

Lemma 2 [3]: Let $\tilde{M} = \mathrm{diag}(\tilde{\mu}_j)$ be a diagonal matrix with $\tilde{\mu}_j > 0$ and $\tilde{\Gamma} = (\tilde{\gamma}_{jk})$ be an $m \times m$ matrix with $\tilde{\gamma}_{jk} \geqq 0$. If $\tilde{M}-\tilde{\Gamma}$ is an M-matrix, there is a diagonal matrix $D = \mathrm{diag}(d_j)$ with $d_j > 0$ such that $\tilde{M}D\tilde{M}-\tilde{\Gamma}^T D\tilde{\Gamma}$ is positive-definite.

Lemma 3: Let F be an $\tilde{n} \times \tilde{n}$ matrix, $\underline{b}$ and $\underline{c}$ $\tilde{n}$-vectors, $\tilde{\xi}$ and $\tilde{\eta}$ scalars satisfying $\tilde{\eta} > \tilde{\varepsilon}$ and Q a positive-definite $\tilde{n} \times \tilde{n}$ matrix. Assume $(F, \underline{b})$ is a completely controllable pair. Then, there exist a positive-definite matrix P and a vector $\underline{q}$ which satisfy

$$F^T P + PF = -\underline{q}\underline{q}^T + \tilde{\xi}\tilde{\eta}\underline{c}\underline{c}^T - \varepsilon Q \tag{28}$$

$$P\underline{b} - \tfrac{1}{2}(\tilde{\xi} + \tilde{\eta})\ \underline{c} = \underline{q} \tag{29}$$

for sufficiently small $\varepsilon > 0$ if

$$h(s) = \frac{1+\tilde{\eta}f(s)}{1+\tilde{\xi}f(s)} \tag{30}$$

is strictly positive real where

$$f(s) = \underline{c}^T (sI-F)^{-1}\underline{b} \ .$$

[Note] If we assume that $F-\tilde{\xi}bc^T$ is Hurwitz, the condition turns out to be necessary, too.

4. PROOFS OF THE MAIN RESULTS

Since Theorems 1 and 2 are nothing but different descriptions of the same fact, we prove them at once. Then, Theorem 3 will be proved.

[Proof of Theorems 1 and 2] Let δ_0 be the minimum eigenvalue of the M-matrix $A-B$ and choose $\delta < \delta_0$. Then, $A-\delta I-B$ remains an M-matrix (Corollary to Th. 6 of [7]). Note that

$$\delta < \delta_0 \leqq \alpha_j \qquad j = 1, \text{---} , m \tag{31}$$

follows from the property of M-matrices. Since $h_j(s)$ is positive real and since (31) implies $\eta_j + \alpha_j - \delta > \xi_j - \alpha_j + \delta$, we obtain that $\tilde{h}_j(s)$ given by

$$\tilde{h}_j(s) = \frac{1+(\eta_j+ \delta_j-\delta)f_j(s)}{1+(\xi_j- \alpha_j+\delta)f_j(s)} \tag{32}$$

is strictly positive real. Therefore, by Lemma 3, there exists P_j which satisfies the requirements of Theorem 2. Let us define $v_j(\underline{x}_j, t)$ by

$$v_j(\underline{x}_j, t) = \underline{x}_j^T P_j \underline{x}_j \tag{33}$$

and apply Lemma 1. For the system CS , the functions $\underline{\tilde{f}}_j$ and $\underline{\tilde{g}}_j$ are given by

$$\underline{\tilde{f}}_j(\underline{x}_j, t) = F_j\underline{x}_j - \underline{b}_j\phi_j(y_j, t)$$

$$\underline{\tilde{g}}_j(\underline{x}_1, \text{---} , \underline{x}_m, t) = \underline{b}_j g_j(y_1, \text{---} , y_m, t)$$

and the isolated subsystem (19) turns out to be

$$\dot{\underline{x}}_j = F_j\underline{x}_j - \underline{b}_j\phi_j(y_j, t) \ , \quad y_j = \underline{c}_j^T\underline{x}_j \ . \tag{34}$$

By (9) and (10), we obtain

$$\dot{v}_j\big|_{(34)} = \underline{x}_j^T(F_j^T P_j + P_j F_j)\underline{x}_j - 2\phi_j(y_j, t)\underline{b}_j^T P_j \underline{x}_j$$

$$= -\underline{x}_j^T \underline{q}_j \underline{q}_j^T \underline{x}_j + (\xi_j - \alpha_j + \delta)(\eta_j + \alpha_j - \delta)\underline{x}_j^T \underline{c}_j \underline{c}_j^T \underline{x}_j$$

$$-\varepsilon_j \underline{x}_j^T \underline{x}_j - 2\phi_j(y_j,\ t)\underline{b}_j^T P_j \underline{x}_j$$

$$= -(\underline{x}_j^T P_j \underline{b}_j)^2 - 2(\underline{x}_j^T P_j \underline{b}_j)[\phi_j(y_j,\ t) - \tfrac{1}{2}(\eta_j + \varepsilon_j)\underline{c}_j^T \underline{x}_j]$$

$$- \tfrac{1}{4}(\eta_j - \varepsilon_j + 2\alpha_j - 2\delta)^2 (\underline{c}_j^T \underline{x}_j)^2 - \varepsilon_j \underline{x}_j^T \underline{x}_j \quad . \tag{35}$$

From the assumption (3) we obtain

$$|\phi_j(y_j,\ t) - \tfrac{1}{2}(\eta_j + \varepsilon_j) y_j| \leqq \tfrac{1}{2}(\eta_j - \varepsilon_j)|y_j| \ . \tag{36}$$

Therefore, noting that $y_j = \underline{c}_j^T \underline{x}_j$, we obtain from (35)

$$\dot{v}_j|_{(34)} \leqq -[u_{1j}(\underline{x}_j)]^2 + (\eta_j - \xi_j) u_{1j}(\underline{x}_j) u_{2j}(\underline{x}_j)$$

$$- \tfrac{1}{4}(\eta_j - \xi_j + 2\alpha_j - 2\delta)^2 [u_{2j}(\underline{x}_j)]^2 - u_{0j}(\underline{x}_j) \tag{37}$$

where

$$u_{1j}(\underline{x}_j) = |\underline{x}_j^T P_j \underline{b}_j| \ , \ \ u_{2j}(\underline{x}_j) = |\underline{c}_j^T \underline{x}_j| \ , \ \ u_{0j}(\underline{x}_j) = \varepsilon_j \underline{x}_j^T \underline{x}_j \ . \tag{38}$$

As for the interconnection, we obtain the next from (4) and (33)

$$\left|[\mathrm{grad}_j v_j]^T \underline{b}_j g_j(y_1,\ ---,\ y_m,\ t)\right| \leq 2|\underline{x}_j^T P_j \underline{b}_j| \sum_{k=1}^{m} \beta_{jk}|y_k|$$

$$= 2u_{1j}(\underline{x}_j) \sum_{k=1}^{m} \beta_{jk} u_{2k}(\underline{x}_k) \ . \tag{39}$$

Thus, the requirements (a) and (b) of Lemma 1 are satisfied with the values of the constants κ_j, λ_j, μ_j and γ_{jk} being

$$\kappa_j = 1,\ \lambda_j = \tfrac{1}{2}(\eta_j - \varepsilon_j),\ \mu_j = \tfrac{1}{4}(\eta_j - \varepsilon_j + 2\alpha_j - 2\delta)^2,\ \gamma_{jk} = \beta_{jk}$$

$$j,\ k = 1,\ ---\ ,\ m \ . \tag{40}$$

So, the test matrix $K^{\frac{1}{2}}M^{\frac{1}{2}} - (\Lambda + \Gamma)$ in the requirement (c) of Lemma 1 is equal to $A - \delta 1 - B$. Since $A - \delta 1 - B$ is an M-matrix from the choice of δ , we can conclude that all the requirements of Lemma 1 are satisfied and, therefore, the origin of the system CS is asymptotically stable. Thus, we have proved Theorem 1 the the existence of P_j in Theorem 2.

The existence of D in Theorem 2 follows, by Lemma 2, from the fact that $\tilde{M} - \tilde{\Gamma} = A - \delta 1 - B$ is an M-matrix. The last part of Theorem 2 is evident from the proof of Lemma 1. [Q.E.D.]

[Proof of Theorem 3] As found in the proof of Theorems 1 and 2 and the proof of Lemma 1, the negative-definiteness of the derivative of $v(\underline{x})$ along the solution of the system CS is guaranteed in the subset of the state space given by

$$-\rho_{1j} \leqq \underline{c}_j^T \underline{x}_j \leqq \rho_{2j} \qquad j = 1,\ ---\ ,\ m \ . \tag{41}$$

Evidently, S given by (13) is included in this subset. Therefore, by the form of S , we can conclude that any solution starting from an initial point contained in S converges to the origin. [Q.E.D.]

5. CONCLUSION

In this paper we obtained:

(i) a proof of the parallelism between Lyapunov stability analysis and the input-output stability analysis of composite systems,

(ii) the frequency-domain condition for local asymptotic stability under assumptions that the linear parts of subsystems may have unstable poles and that the inequalities about the nonlinear functions are satisfied only locally (Theorem 1),

(iii) Lyapunov functions of subsystems which are suited to the composite-system method (Theorem 2),

(iv) an estimate of the domain of attraction (Theorem 3),

(v) and a new stability condition of composite systems which is applicable to a wide class of systems (Lemma 1).

Among the five, the items (ii)-(iv) are important from a practical viewpoint. Actually, a real large-scale system is seldom stable in the large and analysis of local stability, especially calculation of the domain of attraction, is strongly expected in many cases. In such an analysis, construction of the Lyapunov functions for subsystems has been the crucial and most troublesome step in the overall procedure. As a result, various efforts have been made to provide a "good" or "handy" Lyapunov functions of subsystems. For instance, in [8] and in Sec. 6.5 of [9], Lyapunov functions which are "best" in the sense that the admissible bounds for the interconnections become the largest among those obtainable from certain stability theorems are calculated out, and in Sec. 7.3 of [9] a "handy" way to construct Lyapunov functions using factorization of s-functions was shown. Our present result carries both "handiness" and "goodness" to a certain extent. Namely, it is handy in the point that all the values needed to establish the overall stability can be obtained graphically. This fact enables us to obtain a comparatively transparent view over the influence of the parameters on the overall stability. On the other hand, the results of the analysis of multimachine power systems reported in another article [10] show that our present method gives a considerably sharper result than direct application of the previous theorems [9], [2]. The improvement seems to originate from the fact that the present method makes use of the structural peculiarity of the system CS that the interconnections depend only on the subsystems' outputs y_j and are added to the subsystems' inputs u_j .

The items (i) and (v) are of theoretical importance. It is possible to formulate a theorem similar to Lemma 1 which employs more than two sets of functions on the righthand sides of (18) and (20). Our present result can be extended to the case in which u_j and y_j are vectors of the same dimension.

REFERENCES

[1] M. Araki, "Input-output stability of composite feedback systems," IEEE Trans. Automat. Contr., vol. AC-21, pp. 254-259, 1976.

[2] M. Araki, "Stability of large-scale nonlinear systems - quadratic-order theory of composite-system method using M-matrices," IEEE Trans. Automat. Contr., vol. AC-23, pp. 129-142, 1978.

[3] M. Araki, "Application of M-matrices to the stability problems of composite dynamical systems," J. Math. Anal. Appl., vol. 52, pp. 309-321, 1975.

[4] Z.V. Rekasius and J.R. Rowland, "A stability criterion for feedback systems containing a single time-varying nonlinear element, IEEE Trans. Automat. Contr., vol. AC-7, pp. 352-354, 1962.

[5] K.S. Narendra and J.H. Taylor, Frequency Domain Criteria for Absolute Stability, New York: Academic Press, 1973.

[6] D. Siljak and S. Weissenberger, "A construction of the Lur'e-Liapunov function," Regelungstechnik and Prozess-Datenverarbeitung, vol. 10, pp. 455-456, 1970.

[7] M. Araki and B. Kondo, "Stability and transient behavior of composite nonlinear system," IEEE Trans. Automat. Contr. vol. AC-17, pp. 111-116, 1972.

[8] M. Araki and B. Kondo, "A stability condition of large-scale composite systems II," J. Japan Association of Automatic Control Engineers, vol. 18, pp. 249-255, 1974 (in Japanese. The same contents are included in: M. Araki, Stability of Composite Control Systems, Ph.D. Dissertation, Fac. Eng., Kyoto Univ., Kyoto, Japan, 1971).

[9] D. Siljak, Large-Scale Dynamic Systems, New York:, North-Holland, 1978.

[10] M. Araki, M. Saeki and B. Kondo, "Application of a new stability condition of composite systems to multimachine power systems," submitted to IEEE Trans. Automat, Contr.

[11] J. E. Potter, "Matrix quadratic solutions," SIAM J. Applied Math., vol. 14, pp. 496-501, 1966.

THE PADÉ-HURWITZ METHOD OF MODEL REDUCTION

Y. BISTRITZ and U. SHAKED
School of Engineering
Tel-Aviv University
Tel-Aviv
Israel

ABSTRACT

Padé approximations to Hurwitz polynomials are defined and general conditions under which the approximants are Hurwitz are presented. Using these approximations a new Padé-Hurwitz method of model reduction for linear time invariant systems, which is entirely based on Padé equations, is established. The method yields stable reduced order models which approximate the low and the high frequency responses of the given stable high order system and does not require any eigenvalue computations.

INTRODUCTION

Most of the analytical methods that have been proposed for the order reduction of linear time invariant stable systems and avoid computations of eigenvalues do not guarantee that the low order approximating models are also stable [1], [2], [3]. The only exception is the Routh Approximation Method (RAM) which has been developed by [4] and used by others [5], [6]. This method is restricted, however, to approximations about a single (the zero) frequency point and as such may yield, in many cases, unsatisfactory approximations.

The objective of the present paper is to introduce a method which applies the computationally attractive Padé technique to obtain low order approximations that overcome the above limitations.

This method approximates low-frequency, high-frequency or both low-and-high-frequency responses of the given high order system by a unified Padé computational procedure which permits any desired weightings between the slow and the fast responses where the need for such approximations in practical cases has been recognized by [2], [7]. In addition, if the given system is stable, the resulting reduced order models are stable too.

MATHEMATICAL PRELIMINARIES

Padé Approximation (PA), [3].

Let $f(x)$ be any function regular at $x=0$. The rational function $R_{k,\ell}(x)$

$$R_{k,\ell}(x) = \frac{a_0+a_1x+a_2x^2+\ldots+a_kx^k}{b_0+b_1x+b_2x^2+\ldots+b_\ell x^\ell} \tag{1}$$

(which is henceforth termed as a $[k,\ell]$ degree rational function, or simply, a $[k,\ell]$ function) is a $[k,\ell]$PA of $f(x)$ at $x=0$ if all its $k+\ell+1$ degrees of freedom (which, assuming $b_0=1$ are all its coefficients) are determined by th requirement that its Tailor series expansion about $x=0$ has first $k+\ell+1$ terms equal to those of the expansion of $f(x)$ about $x=0$.

Similarly Padé approximation can be defined at any point at which $f(x)$ is regular. Especially, a $[k,\ell]$ PA at $x = \infty$ is defined for $f(x)$, which is regular there, as a $[k,\ell]$ rational function whose coefficients are all determined such that the first $k+\ell+1$ terms in the expansion about $x = \infty$ (expansion in negative powers of x) of the approximant and of $f(x)$ are identical.

Partial Padé Approximation (PPA) [4]

A $[k,\ell]$ rational function that approximates only the first j terms of $f(x)$ in the Pade sense, $j < k+\ell+1$, is called a $[k,\ell]$ j-th PPA.

Modified Padé Approximation (MPA)

Let f(x) be regular at x=0 and x=∞. A [k,ℓ] rational function is called a MPA of f(x) and is denoted by $[k,\ell]^j_i$ if it forms a j-th (j > 0) PPA at x = ∞ and an i-th PPA at x = 0 of f(x), where i+j = k+ℓ+1. Note that there are k+ℓ different [k,ℓ] MPA's, given by $[k,\ell]^j_{k+\ell+1-j}$, j =1,2,...,k+ℓ

Associated Rational Function (ARF)

Given a polynomial

$$D_\nu(s) = \sum_{i=0}^{\nu} d_i s^i \tag{2}$$

The ARF of $D_\nu(s)$ is defined as the [m,n] rational function $r_\nu(x)$ which is derived from the coefficients of $D_n(s)$ by

$$r_\nu(x) = \frac{\sum_{i=0}^{m} d_{2i+1} x^i}{\sum_{i=0}^{n} d_{2i} x^i} \tag{3}$$

where m=n-1 for ν=2n, m=n for ν=2n+1.

The above definitions can be used to define lower degree polynomial approximants according to the following definition.

Associated Pade Approximation (APA)

A polynomial $D_\mu(s)$ is said to be a [k,ℓ] APA of the polynomial $D_\nu(s)$ ($\mu < \nu$) if the ARF of $D_\mu(s)$ is a [k,ℓ] PA of the ARF of $D_\nu(s)$.

PADÉ-HURWITZ POLYNOMIAL APPROXIMATIONS

Given a Hurwitz polynomial $D_\nu(s)$ of degree ν with real positive coefficients, conditions under which APA's of $D_\nu(s)$ are also Hurwitz have been found in [9]. These conditions yield an immediate simple technique to obtain desired lower degree Hurwitz polynomials and are stated here without proofs.

Theorem 1 The [k-1,k] and [k,k] APA's taken at zero of a Hurwitz polynomial $D_\nu(s)$ are Hurwitz for any k = 1,2,...n.

Theorem 2 The [k-1,k] and [k,k] APA's, taken at infinity, of the Hurwitz polynomials $D_{2n}(s)$ and $D_{2n+1}(s)$, respectively, are Hurwitz for any k=1,2,...,n

It is noted that only even (odd) degree Hurwitz APA's of even (odd) Hurwitz polynomials can be obtained because an [n-1,n] ARF, (an [n,n] ARF), has only [k-1,k] ([k,k]), k < n, proper PA's at infinity. (This constraint can be elivated by a slight modification of the ARF definition).

The last theorem concerns the case of Pade-Hurwitz approximations that can be obtained by MPA's taken both at zero and infinity.

Theorem 3 All the combinations of the APA's $[k-1,k]^j_{2k-j}$ (j = 1,2,...,2k-1) and $[k,k]^j_{2k+1-j}$ (j = 1,2,...,2k) of the Hurwitz Polynomials $D_{2n}(s)$ and $D_{2n+1}(s)$, respectively, are Hurwitz for any k = 1,2,...,n.

Here too using APA's which involve approximations at infinity only even (odd) Padé-Hurwitz polynomial approximants can be obtained for a given even (odd) Hurwitz polynomial.

Let $D_\nu(s)$ be a Hurwitz polynomial with the ARF

$$r_\nu(x) = \frac{a_0+a_1x+\ldots+a_Mx^M}{b_0+b_1x+\ldots\ b_Nx^N} \tag{4}$$

where $M = N-1$ for $\nu = 2N$ and $M = N$ for $\nu = 2N+1$ and assume that $b_0=1$.

Let the expansion of $r_\nu(x)$ about $x = 0$ and $x = \infty$ be given by

$$r_\nu(x) = \sum_{i=0}^{\infty} c_ix^i \quad \text{and} \quad r_\nu(x) = \sum_{i=0}^{\infty} m_ix^{-i} \tag{5a.b}$$

respectively, where for $\nu=2N$ $(M = N-1)$ $m_0=0$

Theorems 1-3 can be summarized by stating that the APA $D_\mu(s)$ of the Hurwitz polynomial $D_\nu(s)$ is Hurwitz iff the ARF $r_\mu(x)$ of $D_\mu(s)$ is either

(i) $[n-1,n]_L^J$ PA of $r_\nu(x)$ for $\nu = 2N$ and $\mu = 2n$

or

(ii) $[n,n]_L^J$ PA of $r_\nu(x)$ for $\nu = 2N+1$ and $\mu = 2n+1$

for all $J = 0,1,\ldots\ \nu$ and $n = 1,2,\ldots,N$, where $L = \mu - J$.

Let the approximant $r_\mu(x)$ be written as

$$r_\mu(x) = \frac{\alpha_0+\alpha_1x+\ldots\alpha_mx^m}{\beta_0+\beta_1x+\ldots\ \beta_nx^n} \tag{6}$$

where either $m = n-1$ or $m = n$ and assume that $\beta_0 = 1$. The equations for the coefficients $\{\alpha_i\}_{i=0}^m$ and $\{\beta_i\}_{i=1}^n$ can be derived for each case by fitting coefficients of the first J and L terms of the power series expansions of $r_\nu(x)$ and $r_\mu(x)$ about $x = 0$ and $x = \infty$, respectively. A convenient uniform representation of these equations that includes all the possibilities is the following:

Defining coefficients $\{f_i\}_{i=0}^\infty$ in terms of the coefficients of the expansions about $x = 0$ and $x = \infty$, $\{c_i\}_{i=0}^\infty$ and $\{m_i\}_{i=0}^\infty$, for $J\geq 0$, by

$$f_i = -m_{J-i} \quad i = 0,1,\ldots\ J-1; \quad f_J = -m_0 + c_0; \quad f_{J+i} = c_i \quad i = 1,2,3,\ldots \tag{7}$$

The coefficients $\{\beta_i\}_{i=1}^n$ are obtained by solving the set of n linear equations which as formulated by the special Toeplitz structures

$$\begin{bmatrix} f_{n-1} & f_{n-2} & \cdots & f_1 & f_0 \\ f_n & f_{n-1} & & f_2 & f_1 \\ \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & & \cdot & \cdot \\ f_{2n-1} & f_{2n-2} & \cdots & f_n & f_{n-1} \end{bmatrix} \cdot \begin{bmatrix} \beta_1 \\ \beta_2 \\ \cdot \\ \cdot \\ \cdot \\ \beta_n \end{bmatrix} = \begin{bmatrix} -f_n \\ -f_{n+1} \\ \cdot \\ \cdot \\ \cdot \\ -f_{2n} \end{bmatrix}, \text{for } [n-1,n]_L^J \tag{8a}$$

and

$$\begin{bmatrix} f_n & f_{n-1} & \dots & f_2 & f_1 \\ f_{n+1} & f_n & \dots & f_3 & f_2 \\ \cdot & & & & \\ \cdot & & & & \\ \cdot & & & & \\ f_{2n} & f_{2n-1} & \dots & f_{n+1} & f_n \end{bmatrix} \cdot \begin{bmatrix} \beta_1 \\ \beta_2 \\ \cdot \\ \cdot \\ \cdot \\ \beta_n \end{bmatrix} = \begin{bmatrix} -f_{n+1} \\ -f_{n+2} \\ \cdot \\ \cdot \\ \cdot \\ -f_{2n+1} \end{bmatrix}, \text{ for } [n,n]_L^J \tag{8b}$$

The coefficients $\{\alpha_i\}_{i=0}^m$ are then readily found by

$$\alpha_k = \begin{cases} \sum\limits_{i=0}^{k} c_i \beta_{k-1} & J \le L \\ & \qquad k = 0,1,\dots,m \\ \sum\limits_{i=k}^{m} m_{i-k} \beta_i & J \ge L \end{cases} \tag{9}$$

Algorithms for simplified digital solution of such equations may be used [10]. Note that for the model reduction application represented in the next sections, μ is the order of the reduced model and the number of equations to be solved, $n=[\mu/2]$, is very small for many practical situations (and may readily obtained in many cases without machine calculations).

THE PADE HURWITZ MODEL REDUCTION METHOD

Let the transfer function G(s) of a linear time invariant system of order ν be

$$G(s) = \frac{N(s)}{D_\nu(s)} = \frac{n_0+n_1s+ \quad +\dots n_\mu s^\mu}{d_0+d_1s+ \quad +\dots d_\nu s^\nu} \qquad \mu < \nu \tag{10}$$

The model reduction problem is to find a reduced $\hat{\nu}$-th order transfer function $\hat{G}(s)$ (usually $\hat{\nu} < < \nu$)

$$\hat{G}(s) = \frac{\hat{N}(s)}{D_{\hat{\nu}}(s)} = \frac{\hat{n}_0+\hat{n}_1s+\dots+\hat{n}_{\hat{\mu}}s^{\hat{\mu}}}{\hat{d}_0+\hat{d}_1s+\dots \quad \hat{d}_{\hat{\nu}}s^{\hat{\nu}}} \qquad \hat{\mu} \le \hat{\nu} \tag{11}$$

that approximates G(s) in some desired sense.

The theory developed in the previous sections can be applied to form a model reduction method which resolves the stability problem on one hand and keeps the desired simplicity of the Padé approximation procedure on the other.

The Padé-Hurwitz method of model reduction which depends entirely on the simple Padé technique is obtained by three stages. The various necessary series expansions are first derived. Then the denominator $D_{\hat{\nu}}(s)$ is obtained as a $\hat{\nu}$-th degree APA of $D_\nu(s)$ that tends to approximate poles near at and far from the origin (in numbers determined by the parameters L and J, respectively, of the APA). Finally the numerator $\hat{N}(s)$ is calculated by requiring $\hat{G}(s)$ to be a $[\hat{\mu},\hat{\nu}]$ $\hat{\mu}$-th PPA of G(s) at zero or infinity.

Expansions in Power series

Assuming, without loss of generality, that $d_0=1$, the function G(s) of (10) can be expanded about zero and infinity by

$$G(s) = \sum_{i=0}^{\infty} \tau_i s^i \qquad \text{and} \qquad G(s) = \sum_{i=0}^{\infty} \mu_i s^{-i} \tag{12a,b}$$

respectively, where

$$\tau_0 = n_0, \ \mu_0 = \frac{n_\nu}{d_\nu}, \quad \tau_i = n_i - \sum_{k=1}^{i} d_i \tau_{i-k}, \quad \mu_i = \frac{1}{d_\nu} [n_{\nu-i} - \sum_{k=1}^{i} d_{\nu-k}\mu_{k-1}], \quad i=1,2,\ldots \tag{13}$$

The coefficients τ_i are proportional to the time moments of the system (9) and the coefficnets μ_i are called the Markov parameters of this system [2]

In both expansions it is understood that $n_i = 0$ for $i > \mu$, $d_i = 0$ for $i > \nu$ and that, given $\mu < \nu$, at least $\mu_0 = 0$ (the first $\nu - \mu$ terms in (12) are zero).

Let the ARF of the denominator, $D_\nu(s)$, of $G(s)$ be

$$r_\nu(x) = \frac{a_0 + a_1 x + \ldots . a_M x^M}{b_0 + b_1 x + \ldots . b_N x^N} \tag{14}$$

where clearly $b_i = d_{2i}$, $a_i = d_{2i+1}$, $M = N-1$ for $\nu = 2N$, $M = N$ for $\nu = 2N+1$ and $b_0 = 1$ by assumption. $r_\nu(x)$ can be expanded about $x = 0$ and $x = \infty$ by (5a,b), where

$$c_0 = a_0, \ m_0 = \frac{a_N}{b_N}, \ c_i = a_i - \sum_{k=1}^{i} b_k c_{k-i}, \ m_i = \frac{1}{b_N} [a_{N-i} - \sum_{k=1}^{i} b_{n-k} m_{k-1}] \ i=1,2,\ldots \tag{15a,b}$$

and where in both expansions of $r_\nu(x)$, $a_i = 0$ for $i > M$, $b_i = 0$ for $i > N$ and in (15b) $a_N = m_0 = 0$ for the case $M = N-1$ ($\nu = 2N$).

Obtaining the reduced model denominator

The reduced degree denominator $D_{\hat\nu}(s)$ of a $\hat\nu$-th reduced order system is obtained either by $[n-1,n]_L^J$ APA, $\hat\nu = 2n$ for $\nu = 2N$, or by $[n,n]_L^J$ APA, $\hat\nu = 2n+1$ for $\nu = 2N+1$, for any $J = 0,1,\ldots,\hat\nu$ and $L = \hat\nu - J$

Let the ARF of $D_{\hat\nu}(s)$ be written by

$$r_{\hat\nu}(x) = \frac{\alpha_0 + \alpha_1 x + \ldots . \alpha_m x^m}{\beta_0 + \beta_1 x + \ldots . \beta_n x^n} \tag{16}$$

where $\beta_0 = 1$ and either $m = n-1$ for $\hat\nu = 2n$ or $m = n$ for $\hat\nu = 2n+1$.

The $[m,n]_L^J$ APA $D_{\hat\nu}(s)$ of $D_\nu(s)$ is obtained by finding $r_{\hat\nu}(x)$, the $[m,n]_L^J$ PA of $r_\nu(x)$ applying the following steps;

(i) Construct from c_i and m_i of (15) the coefficients f_i of (7).

(ii) Solve the Toeplitz structure equations (8) for $\beta_1, \beta_2, \ldots, \beta_n$

(iii) Compute $\alpha_0, \alpha_1, \ldots \alpha_m$, using (9)

(iv) The coefficient of $D_{\hat\nu}(s)$ are then given by

$$\hat{d}_{2i} = \beta_i \quad i = 0,1,\ldots,n \;;\; \hat{d}_{2i+1} = \alpha_i \quad i = 0,1,\ldots,m \tag{17}$$

For a given stable system the stability of the model whose characteristic polynomial is $D_{\hat{\nu}}(s)$ is assured by the theorems 1-3. The model obtained tends to possess L poles close to and J poles far from the origin.

Obtaining the reduced model numerator

The numerator polynomial $\hat{N}(s)$ is determined by requiring $\hat{G}(s)$ to be a $[\hat{\nu}, \hat{\nu}]$ λ-th PPA at $s = 0$ and a $(\hat{\nu}-\lambda)$-th PPA of $G(s)$ at $s = \infty$ (λ is an integer $1 \leq \lambda \leq \hat{\nu}$)

For a λ-th PPA at $s = 0$, $\hat{G}(s)$ retains λ of the first time moments of $G(s)$. Using λ terms in the expansion (12a) the first λ numerator coefficients are determined by

$$\hat{n}_i = \sum_{k=0}^{i} \hat{d}_k \tau_{i-k} \qquad i = 0,1,\ldots, \lambda-1 \tag{18a}$$

The rest of the numerator coefficients are determined by fitting the first Markov parameters of $\hat{G}(s)$ found in (12b)

$$\hat{n}_i = \sum_{k=0}^{\hat{\nu}-i} \hat{d}_{\hat{\nu}-k} \mu_{\hat{\nu}-i-k} \qquad i = \lambda,\lambda+1,\ldots,\hat{\nu} \tag{18b}$$

The parameter λ determines the amount by which $\hat{G}(s)$ tends to approximate
the steady state and the transient response of $G(s)$, [9],[5], where $\lambda = \hat{\nu}$ ($\lambda=1$) corresponds to steady state (transient) behaviour approximation.

The final response of a reduced $\hat{\nu}$-th order model is determined by its stability and by the two parameters L and λ ($L = \hat{\nu}-J$)

Illustrative example

Consider the 8-th order system with a transfer function $G(s)=N(s)/D_8(s)$ whose numerator and denominator are

$$N(s) = 971.8333 \cdot \{ 3\cdot10^6 + 5.41\cdot10^6 s+3.117\cdot10^6 s^2+7.937\cdot10^5 s^3+9.147\cdot10^4 s^4+ + 4.886\cdot10^3 s^5+1.17\cdot10^2 s^6+s^7\} \tag{19}$$

$$D_8(s) = 2.9155\cdot10^9 + 3.22567\cdot10^9 s + 1.3934708\cdot10^9 s^2 + 2.5991948\cdot10^8 s^3 + + 2.2990255\cdot10^7 s^4 + 1.021512\cdot10^6 s^5 + 2.2962\cdot10^4 s^6 + 2.4510^2 s^7 + s^8$$

G(s) has the following zeros and poles all of which are in the left half plane:

zeros; -1., -3.±j1., -10., -20., -30., -50.

Poles; -2±j1., -7., -14., -25., -35., -80±j20.

These are given for demonstration purposes only as in the Padé-Hurwitz method their calculation is not required.

Reduced models of order 1,2,3,4 by Padé approximating G(s) at s=0 directly have been calculated and have been found to lead to unstable models.

Assuming that a 4-th order reduced model is required the Padé-Hurwitz method suggests five different stable reduced models corresponding to the different possible choices of the J parameter J = 0,1,2,3,4 (for λ-1= L = 4-J). This parameter determines the amount by which the transient response (far off poles and Markov parameters) is approximated. Following the theory developed above these five models have been derived and their zeros and poles summarized in Table II. In order to demonstrate the algorithm one of these models (J=1) is now calculated explicitly.

By (a) the ARF of $D_8(s)$ is

Table I

Coefficient of the expansions about zero of and infinity for the example.

i	$G(s) = \sum_{i=0}^{\infty} \tau_i s^i$ τ_i	$G(s) = \sum_{i=0}^{\infty} \mu_i s^{-i}$ μ_i	$r(x) = \sum_{i=0}^{\infty} c_i x^i$ c_i	$r(x) = \sum_{i=0}^{\infty} m_i x^{-i}$ m_i
0	1.	0	1.1064	0
1	$6.9694 \cdot 10^{-1}$	$9.7183 \cdot 10^{2}$	$-4.3965 \cdot 10^{-1}$	$2.4500 \cdot 10^{2}$
2	$-2.1004 \cdot 10^{-1}$	$-1.2439 \cdot 10^{5}$	$2.0176 \cdot 10^{-1}$	$-4.6042 \cdot 10^{6}$
3	$-7.4699 \cdot 10^{-2}$	$1.2910 \cdot 10^{7}$	$-9.2972 \cdot 10^{-2}$	$1.0035 \cdot 10^{11}$

Table II

4-th order approximants for various values the parameter J (λ-1=L=4-J)

Parameter J	Poles	Zeros
0	-2.0019 ± j0.9977 -5.4143 ± j4615	-0.9515 -1.5452 ± j1.3114
1	-2.8816 -1.9571 ± j1.1632 -238.20	-0.9696 -1.7197 ± j1.2238
2	-1.4103 ± j0.7689 -121.09 ± j60.478	-0.9644 ± j0.3910 -44.919
3	-0.9289 -71.235 ± j45.3218 -101.60	-0.5866 -11.219 -105.19
4	-40.779 ± j23.709 -81.721 ± j19.995	-3.9415 -56.529 ± j30.265

$$r = \frac{3.22567\cdot 10^{9} + 2.5991948\cdot 10^{8}\ x + 1.021512\cdot 10^{6}x^{2} + 2.45\cdot 10^{2}\ x^{3}}{2.9155\cdot 10^{9} + 1.3934708\cdot 10^{9}x + 2.299255\cdot 10^{7}x^{2} + 2.2962\cdot 10^{4}x^{3} + x^{4}} \qquad (20)$$

The four first terms of the expansions of G(s) and r(x) about zero and infinity can be obtained using (13),(15) and are given in Table I (these are enough to obtain any of the 4-th order models).

Denote the ARF of the required model denominator $\hat{D}_4(s)$, by

$$\hat{r}_4(x) = \frac{\alpha_0 + \alpha_1 x}{\beta_0 + \beta_1 x + \beta_2 x^2} \qquad (21)$$

That is

$$\hat{D}_4(s) = \beta_0 + \alpha_0 s + \beta_1 s^2 + \alpha_1 s^3 + \beta_2 s^4 \qquad (22)$$

β_1, β_2 are obtained by solving (8a)

$$f_1\beta_1 + f_0\beta_2 = -f_2$$
$$f_2\beta_1 + f_1\beta_2 = -f_3$$

where the f_i's are defined by (7) and for J=1 are $f_0 = -m_1$, $f_1 = c_0$, $f_2 = c_1$, $f_3 = c_2$ thus,

$\beta_1 = 4.596\cdot 10^{-1}$, $\beta_2 = 2.811\cdot 10^{-4}$ using(a) it is found that $\alpha_0 = c_0\beta_0 = 1.106$ and $\alpha_1 = c_0\beta_1 + c_1\beta_0 = 6.886.10^{-2}$

Substituting in (22), $\hat{D}_4(s)$ for J=1 is

$\hat{D}_4(s) = 1. + 1.106\ s + 4.596\cdot 10^{-2}s^2 + 6.886\cdot 10^{-2}s^3 + 2.811\cdot 10^{-7}s^4$; J=1

The corresponding numerator of the model transfer function is obtained using the PPA of (18) for $\lambda=3$ it is found that $\hat{n}_0 = \beta_0\tau_0 = 1$, $\hat{n}_1 = \beta_0\tau_1 + \alpha_0\tau_0 = 1.803$, $\hat{n}_2 = \beta_0\tau_2 + \alpha_0\tau_1 + \beta_1\tau_0 = 1.021$, $\hat{n}_3 = \beta_0\tau_3 + \alpha_0\tau_2 + \beta_1\tau_1 + \alpha_1\tau_0 = 0.231$ ($\hat{n}_4 = 0$)

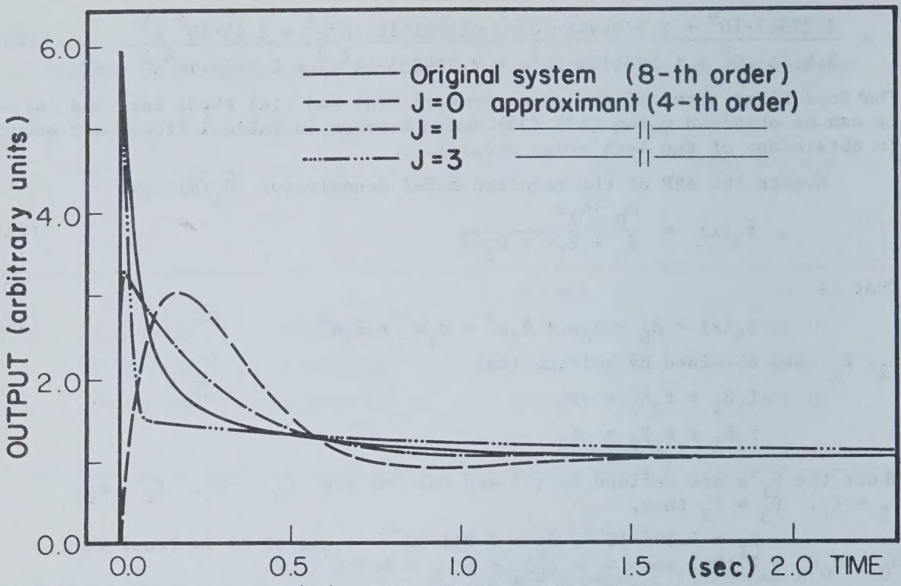
The transfer function of the reduced 4-th order model, for J=1, is therefore

$$G_4(s) = \frac{1 + 1.803\ s + 1.021\ s^2 + 0.231\ s^3}{1 + 1.106\ s + 4.596\cdot 10^{-2}s^2 + 6.886\cdot 10^{-2}s^3 + 2.881\cdot 10^{-4}s^4}$$

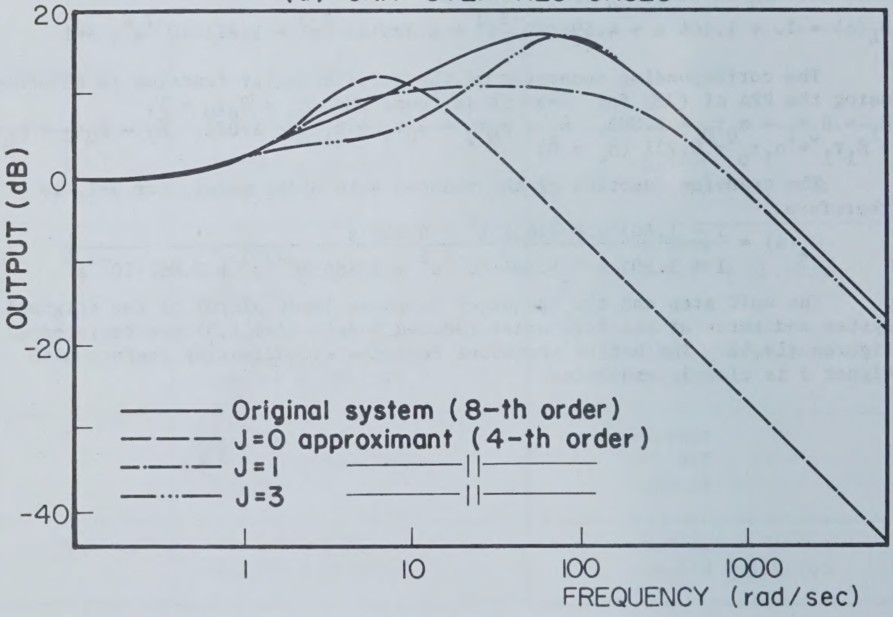
The unit step and the frequency response (Bode plots) of the original system and three of its 4-th order reduced models (J=0,1,3) are depicted in figures (1a,b). The better transient response approximating feature with higher J is clearly exhibited.

Conclusions

A new method for reduced order model approximations of linear time invariant systems is presented. Introducing the concept of the associated Padé approximations to Hurwitz polynomials a method for model reduction which preserves stability and is based entirely on Padé equations is established.



(a) UNIT STEP RESPONSES



(b) BODE PLOTS

Fig. 1: (a) Step responses and (b) Frequency responses for the original system and its fourth-order J=0,1,3 reduced models.

Additional features of the method are

(1) Computationally the method is very simple and suitable for digital calculations. It obviates the need for any eigenvalue evaluations and it is entirely based on the simple Padé equations.

(2) It provides by a single algorithm, different possibilities of approximating a given transfer function at either the low or the high frequency ranges or both. It is shown by an example that this feature can be used to achieve lower order accurate models.

(3) The reduced model has a desired number of eigenvalues near to and far from the origin. The eigenvalues of the $\hat{\nu}$-th order model converge to the exact eigenvalues of the original system with increasing $\hat{\nu}$, [9].

(4) The resulting model fits the first time moments and Markov parameters of the high order system.

(5) The method comprehends in the zero P.A. Case, as it is shown in [9], the existing stability preserving model reduction analytical methods that do not involve eigenvalue computations, (for example [4]-[6]), and provides a simpler algorithm to obtain their results.

References

[1] C.F. Chen and L.S. Shieh 'An algebraic method for control system design' Int. J. Contr. 4, 717-739, 1970.

[2] Y. Shamash 'Linear system reduction using Padé approximation to allow retention of dominant modes' Int. J. Contr., 21,257-272, 1975.

[3] M. Decoster and A.R. Cauwenbergh 'A comparative study of different reduction methods' Part 1 and Part 2, part 1 in Journal A 17, 68-74, 1976, Part 2 in ibid, 125-134, 1976.

[4] M.F. Hutton and B. Friedland 'Routh approximations for reducing order of linear, time invariant systems' IEEE trans. Aut. Contr., AC-20, 324-337, 1975.

[5] Y. Shamash 'Model reduction using the Routh stability criterion and Padé approximation techniques' Int. J. Contr., 21, 475-484, 1975.

[6] V. Krishnamurthi and V. Sehardi 'A simple and direct method of reducing order of linear systems using Routh approximations in the frequency domain' IEEE trans. Aut. Contr., AC-21, 797-799, 1976.

[7] J. Atary and M.M. Shah "Modelling and Analytical Control System Design of a Complete Nuclear Power Plant Prototype" Dept of Engineering, Cambridge University CUED/B Control/TR 16.

[8] G.A. Baker Essentials of Padé Approximations, Academic Press, 1975.

[9] Y. Bistritz and U. Shaked 'Padé approximations to Hurwitz polynomials and their application to linear systems simplification' report TAU/SOE 27/79, 1979 (submitted for publication).

[10] S. Zohar "Toeplitz matrix inversion. The algorithm of F. Trench" J. Assoc. Comp. Mach., 16, 592-601, 1969.

A NOTE ON ORDER-REDUCTION OF LINEAR SYSTEMS VIA PARAMETER OPTIMIZATION

NICOLA SCHIAVONI
Centro di Teoria dei Sistemi
Consiglio Nazionale delle Ricerche
Milano, Italy

ABSTRACT

The problem of synthesizing a low-order model of a high-order system via Parameter Optimization is dealt with in this paper.

First, a set of non-linear matrix algebraic equations, representing necessary optimality conditions for the simplified model, is derived and a Newton-Raphson-type technique is proposed for their solution. Then, by exploiting typical methods of Problem Analysis, a decomposition algorithm is obtained for the set of interconnected linear matrix algebraic equations to be solved at each step of the procedure.

1. INTRODUCTION

Among the techniques proposed to synthesize simplified models of high-order time-invariant linear systems [1]-[3], considerable attention was given to those based on Parameter Optimization.

In this paper, following the approach of [4]-[13], the high-order system as well as its reduced-order model are assumed to have a finite-dimensional state-space representation and the parameters identifying the simplified model are selected so as to minimize a quadratic functional of the approximation error, that is of the difference between the output of the original system and the output of the reduced-order model. Besides, a colored noise with an asymptotically rational spectrum, namely the output of an asymptotically stable linear time-invariant and finite-dimensional shaping filter driven by a white noise, is adopted as test input.

The problem above can be reformulated as a non-linear programming problem, which can be tackled by applying gradient-type algorithms to directly minimize the cost function [7], [9] or, as it will be done in this paper, by imposing that a set of necessary optimality conditions is met [4]-[6], [8], [10]-[13].

When the latter strategy is chosen, a set of non-linear matrix algebraic equations has to be solved and, if one tries to cope with it by means of Newton-Raphson-type techniques, a set of linear matrix algebraic equations, which are interconnected according to a loop-type structure, must be solved at each step of the procedure. This can constitute a non-trivial task as the number of corresponding scalar equations is often very large.

Here special attention is paid to this point and, by exploiting methods of Problem Analysis [14], [15], a decomposition algorithm is developed which allows to solve the whole set of matrix equations by only solving, a finite number of times, the low-dimensional matrix equations constituting the set together with a vector "congruence" equation.

The paper is organized as follows. Section 2 is devoted to a formal statement of the here considered problem and to its recasting in the form of a non-linear programming problem. Then, a set of non-linear matrix algebraic equations, which represent necessary optimality conditions, are derived and discussed in Section 3, whereas Section 4 deals with the aforementioned decomposition algorithm for their solution.

2. PROBLEM STATEMENT AND REFORMULATION

The here adopted formulation of the general problem of determining a low-order model of a high-order continuous-time and time-invariant linear system is as follows:

Problem 1

Given

a) A system to be approximated, described by

$$\dot{x}(t) = Ax(t) + B\,u(t) \tag{1.a}$$
$$y(t) = Cx(t), \tag{1.b}$$

where $u(t) \varepsilon R^m$, $x(t) \varepsilon R^n$, $y(t) \varepsilon R^p$, A, B, C are real matrices of appropriate size and A is Hurwitz;

b) A test input u(.), described by

$$\dot{x}_F(t) = A_F\, x_F(t) + B_F\, w(t) \tag{2.a}$$
$$u\,(t) = C_F\, x_F(t) + D_F\, w(t), \tag{2.b}$$

where $w(t) \varepsilon R^q$, $x_F(t)\, \varepsilon R^{n_F}$, A_F, B_F, C_F, D_F are real matrices of appropriate size, A_F is Hurwitz and w(.) is a zero-mean Gaussian white noise with covariance $W = W' > 0$;

c) The order of the reduced-order model n_R $(<n)$;

Find a reduced-order model of the form

$$\dot{x}_R(t) = A_R\, x_R(t) + B_R\, u(t) \tag{3.a}$$
$$y_R(t) = C_R\, x_R(t), \tag{3.b}$$

where $x_R(t) \varepsilon R^{n_R}$, $y_R(t)\, \varepsilon R^p$, A_R, B_R, C_R are real matrices of appropriate size and A_R is Hurwitz, such as to minimize the performance index

$$J(A_R, B_R, C_R) \triangleq \lim_{t\to\infty} \quad E\{e(t)'\, Q\, e(t)\}, \; Q = Q' > 0, \tag{4}$$

where

$$e(\cdot) \triangleq y(\cdot) - y_R(\cdot) \tag{5}$$

is the approximation error.

Remark 1

No explicit reference to a specific canonical form of system (3) is made in this paper, as unnecessary for the discussion to follow. Of course, giving the state space of the reduced-order system a parsimonious (or minimal, if possible) parametrization is important from a computational point of view.

Remark 2

In many circumstances, it is of interest to obtain a reduced-order model endowed with the same static gain as the original high-order system. The formulation of Problem 1 does not guarantee that this fact occurs. Nevertheless, if the shaping filter(2) is such that the power spectrum of $u(\cdot)$ is essentially located at low-frequency, then the difference between the two static gains may often be acceptable. Alternatively, the constraint

$$C A^{-1} B = C_R A_R^{-1} B_R \tag{6}$$

can be explicitly taken into account. For the sake of simplicity, this will not be done here, even though the subsequent discussion of Problem 1 could be easily extended in order to encompass (6).

$\square$

Now denote by $tr(\cdot)$ the trace operator and consider the following non-linear programming problem:

Problem 2

$$\min_{A_R,\, B_R,\, C_R} (tr\,(B_F W B_F' P_{111}) + tr(2BD_F W B_F' P_{121}) + tr(B_R D_F W(2B_F' P_{13} +$$
$$+ D_F'(2\,B' P_{23} + B_R' P_{33}))))$$

subject to

(i) A_R Hurwitz

(ii) $$A_F' P_{111} + P_{111} A_F + (P_{121} B + P_{13} B_R) C_F + C_F' (P_{121} B + P_{13} B_R)' = 0 \tag{7.a}$$

$$A_F' P_{121} + P_{121} A + C_F' B_R' P_{23}' = 0 \tag{7.b}$$

$$A_F' P_{13} + P_{13} A_R + C_F' (B' P_{23} + B_R' P_{33}) = 0 \tag{7.c}$$

$$A' P_{23} + P_{23} A_R - C' Q C_R = 0 \tag{7.d}$$

$$A_R' P_{33} + P_{33} A_R + C_R' Q C_R = 0 \tag{7.e}$$

$\square$

The relationship between Problem 1 and Problem 2 is specified by the following result (see also [4]-[13]):

Theorem 1

Any reduced-order system (3) solving Problem 1 specifies a triplet (A_R, B_R, C_R) solving Problem 2 and any triplet (A_R, B_R, C_R) solving Problem 2 specifies a reduced-order system (3) solving Problem 1.

Proof

Letting $z'(t) \triangleq \left| x_F(t)'\ x(t)'\ x_R(t)' \right|$,

$$F \triangleq \begin{vmatrix} A_F & 0 & 0 \\ BC_F & A & 0 \\ B_R C_F & 0 & A_R \end{vmatrix},\ G \triangleq \begin{vmatrix} B_F \\ BD_F \\ B_R D_F \end{vmatrix},\ H' \triangleq \begin{vmatrix} 0 \\ C' \\ -C_R' \end{vmatrix}, \tag{8}$$

eqs. (1)-(3), (5) can be rewritten as $\dot{z}(t) = Fz(t) + G\,u(t)$, $e(t) = Hz(t)$, whereas (e.g., see [16]) the performance index (4), provided that A_R (and hence F) is Hurwitz, can be evaluated as $tr(G\ W\ G'P)$ where P solves the Liapounov equation

$$F'P + PF + H'QH = 0. \tag{9}$$

If the partitioning of the (symmetric) matrix P corresponding to the one of F,

$$P = \begin{vmatrix} P_{11} & P_{12} & P_{13} \\ P_{12}' & P_{22} & P_{23} \\ P_{13}' & P_{23}' & P_{33} \end{vmatrix},$$

is introduced and (8) is recalled, it turns out that

$$\begin{aligned} tr(G\ W\ G'P) &= tr(B_F W B_F' P_{11}) + tr(BD_F W(2B_F'\ P_{12} + D_F'\ B'P_{22})) + \\ &+ tr(B_R D_F W(2B_F' P_{13} + D_F'(2B'P_{23} + B_R'\ P_{33}))), \end{aligned} \tag{10}$$

whereas eq. (9) is equivalent to (7.c)-(7.e) and

$$A_F'\ P_{11} + P_{11}\ A_F + (P_{12}B + P_{13}B_R)C_F + C_F'(P_{12}B + P_{13}\ B_R) = 0 \tag{11.a}$$

$$A_F'\ P_{12} + P_{12}A + C_F'(B'\ P_{22} + B_R'\ P_{23}) = 0 \tag{11.b}$$

$$A'\ P_{22} + P_{22}A + C'QC = 0. \tag{11.c}$$

Noting that P_{22} (see eq. (11.c)) and all the matrices which multiply it in eqs. (10)-(11.b) do not depend upon A_R, B_R, C_R, eq. (11.c) can be dropped out and eqs. (7.a), (7.b), substituted to eqs. (11.a), (11.b), respectively. Thus the performance index of Problem 2 differs from J only because of a constant term.

Remark 3

An order-reduction problem which, though different from Problem 1, admit Problem 2 as a reformulation, can be stated in a completely deterministic framework. Specifically, with reference to (1)-(3), it must be assumed that $x(0) = 0$, $x_F(0) = 0$, $x_R(0) = 0$, whereas the performance index (4) has to be substituted by

$$\sum_{i=1}^{m} \int_0^\infty (y^i(t)-y_R^i(t))'\ Q\ (y^i(t)-y_R^i(t))dt,\ Q = Q' > 0,$$

$y^i(\cdot)$ and $y_R^i(\cdot)$, $i = 1, 2, \ldots, m$, denoting the outputs of systems (1) and (3), respectively, correspondent to $w(\cdot) \triangleq w^i\ imp\ (\cdot), w^i \varepsilon R^m$, $i = 1, 2, \ldots, m$, where $imp(\cdot)$ is the Dirac impulse function and $\det \mid w^1, w^2, \ldots, w^m \mid \neq 0$. Then $W \triangleq \sum_{i=1}^{m} w^i w^{i'}$ turns out to be symmetric and positive definite.

3. NECESSARY OPTIMALITY CONDITIONS

The aim of this Section is to derive and discuss a set of necessary optimality conditions for Problem 2 (see also [4]-[6], [8], [10]-[13]).

Theorem 2

If the triplet (A_R, B_R, C_R) solves Problem 2, then there exist matrices $L_{11}, L_{12}, L_{13}, L_{23}, L_{331}, P_{13}, P_{23}, P_{33}$ such that:

$$A_F L_{11} + L_{11} A'_F + B_F W B'_F = 0 \tag{12.a}$$

$$A L_{12} + L_{12} A'_F + B(D_F W B'_F + C_F L_{11}) = 0 \tag{12.b}$$

$$A_R L_{13} + L_{13} A'_F + B_R(D_F W B'_F + C_F L_{11}) = 0 \tag{12.c}$$

$$A_R L_{23} + L_{23} A' + B_R D_F W D'_F B' + B_R C_F L'_{12} + L_{13} C'_F B' = 0 \tag{12.d}$$

$$A_R L_{331} + L_{331} A'_R + B_R D_F W D'_F B'_R + L_{13} C'_F B'_R + B_R C_F L_{13}' = 0 \tag{12.e}$$

$$A'_F P_{13} + P_{13} A_R + C'_F (B' P_{23} + B'_R P_{33}) = 0 \tag{12.f}$$

$$A' P_{23} + P_{23} A_R - C' Q C_R = 0 \tag{12.g}$$

$$A'_R P_{33} + P_{33} A_R + C'_R Q C_R = 0 \tag{12.h}$$

$$L_{13} P_{13} + L_{23} P_{23} + L_{331} P_{33} = 0 \tag{12.i}$$

$$D_F W (B'_F P_{13} + D'_F (B' P_{23} + B'_R P_{33})) + C_F (L_{11} P_{13} + L_{12}' P_{23} + L_{13}' P_{33}) = 0 \tag{12.j}$$

$$C L'_{23} - C_R L_{331} = 0. \tag{12.k}$$

Proof

Consider the following lagrangean function associated to Problem 2:

$$\begin{aligned} L \triangleq{} & \mathrm{tr}(B_F W B'_F P_{111}) + \mathrm{tr}(2 B D_F W B'_F P_{121}) + \mathrm{tr}(B_R D_F W (2 B'_F P_{13} + \\ & + D'_F (2 B' P_{23} + B_R' P_{33}))) + \mathrm{tr}(L_{11} (A'_F P_{111} + P_{111} A_F + (P_{121} B + P_{13} B_R) C_F + \\ & + C'_F (P_{121} B + P_{13} B_R)')) + \mathrm{tr}(2 L_{12} (A'_F P_{121} + P_{121} A + C'_F B'_R P_{23}')) + \\ & + \mathrm{tr}(2 L_{13} (A'_F P_{13} + P_{13} A_R + C'_F (B' P_{23} + B'_R P_{33})) + 2 L_{23} (A' P_{23} + \\ & + P_{23} A_R - C' Q C_R) + L_{33} (A'_R P_{33} + P_{33} A_R + C'_R Q C_R)). \end{aligned}$$

Equating to zero the partial derivatives of L with respect to P_{111}, P_{121}, P_{13}, P_{23}, P_{33}, L_{11}, L_{12}, L_{13}, L_{23}, L_{33}, A_R, B_R, C_R and noting that matrices P_{33} and L_{11} are symmetric, eqs. (12.a)-(12.d), (12.f)-(12.h), (12.j) are easily obtained together with

$$A_R L_{33} + L_{33} A'_R + B_R D_F W D'_F B'_R + 2 L_{13} C'_F B'_R = 0 \tag{13.a}$$

$$2 L_{13} P_{13} + 2 L_{23} P_{23} + (L_{33} + L'_{33}) P_{33} = 0 \tag{13.b}$$

$$2 QC L'_{23} - Q C_R (L_{33} + L'_{33}) = 0 \tag{13.c}$$

$$\begin{aligned} & A'_F P_{111} + P_{111} A_F + (P_{121} B + P_{13} B_R) C_F + \\ & \qquad + C'_F (P_{121} B + P_{13} B_R)' = 0 \end{aligned} \tag{13.d}$$

$$A'_F P_{121} + P_{121} A + C'_F B'_R P'_{23} = 0. \tag{13.e}$$

Matrix L_{33} enters eqs. (13.b), (13.c) only through its symmetric part $L_{331} \triangleq 1/2\,(L_{33} + L'_{33})$. Then eqs. (13.b), (13.c), recalling also that Q is non-singular, are equivalent to eqs. (12.i) and (12.k), respectively, whereas eq. (12.e) follows by adding to eq. (13.a) its transpose.

At last, in view of the stability assumption of (1) and (2), eqs.

(13.d), (13.e) are always (uniquely) solvable with respect to P_{111} and P_{121} (e.g., see [17]), which, on the other hand, do not enter eqs. (12). Therefore, eqs. (13.d)-(13.e) do not express any significant condition to be satisfied by the solutions of Problem 2.

Remark 4

Reduced-order systems satisfying the conditions of Theorem 2 are not necessarily asymptotically stable.

□

Theorem 2 can be exploited in order to find triplets (A_R, B_R, C_R) candidate to solve Problem 2. This calls for the solution of the set of non-linear matrix equations (12).

As a matter of fact, eqs. (12.a), (12.b) are linear in L_{11} and L_{12}, do not depend upon other unknowns and, in view of the stability assumption of (1) and (2), have always one and only one solution (e.g., see [17]). Moreover, they can be solved sequentially (first eq. (12.a) for L_{11} and then eq. (12.b) for L_{12}). Therefore, from now on, L_{11} and L_{12} will be considered as known.

The remaining nine matrix equations are actually non-linear. They can be tackled by means of a Newton-Raphson-type iterative procedure essentially calling for the solution of a (theoretically infinite) sequence of equations obtained by linearizing eqs. (12.c)-(12.k) around their current solution. To be specific, denoting by X^i the current value of any one of the unknown matrices at the beginning of the i-th step of the algorithm, the value assumed at the beginning of the succeeding step is $X^{i+1} = X^i + \delta X^i$, the variations δX^i solving

$$A_R^i \, \delta L_{13}^i + \delta L_{13}^i \, A_F' + f_1^i \, (\delta A_R^i, \delta B_R^i) = 0 \qquad (14.a)$$

$$A_R^i \, \delta L_{23}^i + \delta L_{23}^i \, A' + f_2^i \, (\delta A_R^i . \delta B_R^i, \delta L_{13}^i) = 0 \qquad (14.b)$$

$$A_R^i \, \delta L_{331}^i + \delta L_{331}^i \, A_R^{i'} + f_3^i \, (\delta A_R^i, \delta B_R^i, \delta L_{13}^i) = 0 \qquad (14.c)$$

$$A_F^i \, \delta P_{13}^i + \delta P_{13}^i \, A_R^i + f_4^i \, (\delta A_R^i, \delta B_R^i, \delta P_{23}^i, \delta P_{33}^i) = 0 \qquad (14.d)$$

$$A' \, \delta P_{23}^i + \delta P_{23}^i \, A_R^i + f_5^i \, (\delta A_R^i, \delta C_R^i) = 0 \qquad (14.e)$$

$$A_R^{i'} \, \delta P_{33}^i + \delta P_{33}^i \, A_R^i + f_6^i \, (\delta A_R^i, \delta C_R^i) = 0 \qquad (14.f)$$

$$f_7^i \, (\delta L_{13}^i, \delta L_{23}^i, \delta L_{331}^i, \delta P_{13}^i ; \delta P_{23}^i, \delta P_{33}^i) = 0 \qquad (14.g)$$

$$f_8^i \, (\delta B_R^i, \delta L_{13}^i, \delta P_{13}^i, \delta P_{23}^i, \delta P_{33}^i) = 0 \qquad (14.h)$$

$$f_9^i \, (\delta C_R^i, \delta L_{23}^i, \delta L_{331}^i) = 0, \qquad (14.i)$$

where f_j^i, $j = 1,2,\ldots, 9$, are affine functions of their respective arguments not reported here for the sake of conciseness.

Of course, the procedure is stopped when the variations δX^i, measured according to some suitably chosen norm, can be considered sufficiently small.

4. A DECOMPOSITION ALGORITHM

The discussion of the preceding Section allows to draw the conclusion that the core of the problem of finding triplets (A_R, B_R, C_R) candidate to solve Problem 2 can simply be expressed as follows:

Problem 3

Solve equations (14).

$\square$

Matrix eqs. (14), which are interconnected among them according to a loop-type structure, are equivalent to $N \triangleq n_R(2\,n + 2\,n_F + 2\,n_R + m + p + 1)$ scalar equations. As this number is often very large, solving Problem 3 by any direct method, implying to handle a matrix with N^2 entries, turns out to be an all but trivial task. Aim of this Section is just to develop a decomposition algorithm which, taking into account the structure of the interconnections among eqs. (14), allows to solve Problem 3 avoiding the necessity of handling "large-size" matrices. The above-mentioned algorithm has its roots in some typical methods of Problem Analysis.

The following will be assumed from now on:

Assumption 1

Let $\{\lambda_{F1}, \lambda_{F2}, \ldots \lambda_{Fn_F}\}, \{\lambda_1, \lambda_2, \ldots, \lambda_n\}, \{\lambda_{R1}, \lambda_{R2}, \ldots \lambda_{Rn_R}\}$ denote the spectra of A_F, A, A_R, respectively. Then

i) $\lambda^i_{Rj} + \lambda_{Fk} \neq 0 \quad , \forall j = 1,2, \ldots, n_R \, , \forall k = 1, 2, \ldots, n_F$

ii) $\lambda^i_{Rj} + \lambda_k \neq 0 \quad , \forall j = 1,2, \ldots, n_R \, , \forall k = 1, 2, \ldots, n$

iii) $\lambda^i_{Rj} + \lambda^i_{Rk} \neq 0 \quad , \forall j = 1,2, \ldots, n_R \, , \forall k = 1, 2, \ldots, n_R$.

Remark 5

In view of the asymptotic stability of (1) and (2), Assumption 1 is met if A^i_R is Hurwitz.

$\square$

First, a new problem is associated to Problem 3:

Problem 4

Given $(\delta A^i_R, \delta B^i_R, \delta C^i_R)$, compute (f^i_7, f^i_8, f^i_9) .

$\square$

In view of Assumption 1, Problem 4 has always one and only one solution (e.g., see [17]). Moreover, it can be easily computed by solving, one at a time, eqs. (14.a), (14.e), (14.f) for δL^i_{13}, δP^i_{23}, δP^i_{33}, respectively, and then eqs. (14.b) - (14.d) for δL^i_{23}, δL^i_{331}, δP^i_{13}, respectively. Lastly, f^i_7, f^i_8, f^i_9 are easily evaluated.

It is now apparent that Problem 3 can be considered as solved once the solution of the following "Inverse Problem" [15] of Problem 4 is known:

Problem 5

Find $(\delta A^i_R, \delta B^i_R, \delta C^i_R)$ such that $f^i_7 = 0$, $f^i_8 = 0$, $f^i_9 = 0$.

$\square$

Problem 5 can be tackled by an algorithm calling for no much more than a repeated solution of the "simple" Problem 4; it consists of 2 steps [14], [15]:

i) "Identification" of the relationship between the triplet $(\delta A_R^i, \delta B_R^i, \delta C_R^1)$ and the triplet (f_7^i, f_8^i, f_9^i) ;

ii) Solution of a "congruence equation".

As for the first step, because of the linearity of equations involved, it is a priori known that the triplet (f_7^i, f_8^i, f_9^i) is an affine function of the triplet $(\delta A_R^i, \delta B_r^i, \delta C_R^i)$. Thus, letting

$$d \triangleq |[\delta A_R^i]_1 \; [\delta A_R^i]_2 \; \dots \; [\delta B_R^i]_1 \; [\delta B_R^i]_2 \; \dots \; [\delta C_R^i]_1 \; [\delta C_R^i]_2 \; \dots |' \in R^{n_R(n_R+m+p)}$$

$$s \triangleq |[f_7^i]_1 \; [f_7^i]_2 \; \dots \; [f_8^i]_1 \; [f_8^i]_2 \; \dots \; [f_9^i]_1 \; [f_9^i]_2 \; \dots |' \in R^{n_R(n_R+m+p)}$$

where $[A]_j$ denote the j-th row of any matrix A , it turns out that,

$$s = M\,d + \mu \;, \quad M \in R^{n_R(n_R+m+p) \times n_R(n_R+m+p)} \;, \quad \mu \in R^{n_R(n_R+m+p)} \;.$$

The matrix M and the vector μ are readily identified by solving $n_R(n_R+m+p)+1$ times Problem 4: (a) for $d = 0$, s is equal to μ, whereas, (b) setting d equal to the j-th column of the identity matrix, $s-\mu$ turns out to be the j-th column of M .

The second step calls for the solution of the congruence equation

$$0 = M\,\bar{d} + \mu \;. \tag{15}$$

Observe that, since, under Assumption 1, Problem 4 has one and only one solution, M is invertible whenever Problem 5 has one and only one solution. In this last case, it is specified by the triplet $(\overline{\delta A_R^i}, \overline{\delta B_R^i}, \overline{\delta C_R^i})$ correspondent to $\bar{d}$.

This algorithm for Problem 5 directly leads to a decomposition algorithm for Problem 3:

Decomposition Algorithm

i) Solve $n_R(n_R+m+p)+1$ times Problem 4 to compute M and μ (point (a) and point (b) above),

ii) Solve eq. (15) for $\bar{d}$,

iii) Solve eqs. (14.a) - (14.f) with $\delta A_R^i = \overline{\delta A_R^i}$, $\delta B_R^i = \overline{\delta B_R^i}$, $\delta C_R^i = \overline{\delta C_R^i}$. ☐

The effectiveness property of the Decomposition Algorithm is qualified in the following theorem, whose proof is omitted as trivial in view of the preceding discussion on the algorithm for Problem 5.

Theorem 3

Assume that Problem 3 has one and only one solution.

If Assumption 1 holds, it can be found by means of the Decomposition Algorithm.

□

Some comments on the main features of the proposed algorithm are now in order.

The maximum number of scalar equations to be solved at a time is $N_D \triangleq \max \{n_R \cdot n, \quad n_R \cdot n_F, n_R \cdot (n_R+m+p)\}$, the last figure referring to the congruence equation (15) Of course, $N_D < N$.

The matrices to be handled in the Decomposition Algorithm have a number of elements which, if $m \leq n$, $p \leq n$, does not exceed $\overline{N} \triangleq \max \{n^2, n_f^2, n_r^2(n_R+m+p)^2\}$. Besides comparing this number with N^2, observe that in many cases $\overline{N}$ turns out to be strictly less than N_D^2 : for instance, by assuming $n_F < n$, this happens whenever $n > n_R+m+p$ and $n_R > 1$.

As a matter of fact, the above expression for $\overline{N}$ is correct if eqs. (14.a) - (14.f) are solved by means of an algorithm which, exploiting their structure, do not imply to expand them in vector equations. Of special interest, here, is the algorithm presented in [18] and extended in [19]. In fact, as the computational effort it calls for is to a significant extent required to put the coefficient matrices in Real Schur Form, it is particularly time-saving when several equations with identical coefficient matrices and different known terms have to be solved. In the present case, the Real Schur Forms of A and A_F are determined once at the beginning of the Newton-Raphson procedure so that, at i-th step, only the one of A_R^i must be computed.

Lastly, the Real Schur Form of A_R^i exhibits on the diagonal the real parts of its eigenvalues, so that Assumption 1 and, when eqs. (12) are solved, asymptotic stability of the reduced-order system can be immediately checked.

ACKNOWLEDGEMENT

This paper has been completed while the author was with the Department of Mechanical Engineering, University of California, Berkeley, on leave from Italy, thanks to a NATO fellowship. He is grateful, in a special way to Prof. G. Leitmann, for the support received and for the kind hospitality.

The author thanks his colleagues for many stimulating discussions on order-reduction problem. He is particularly indebted with Prof. G. Guardabassi and A. Locatelli for their technical advice as well as for their continuous encouragement.

REFERENCES

1. Genesio, R., Milanese, M.: "A Note on the Derivation and Use of Reduced Order Models", IEEE Trans. Automat. Contr., Vol. AC-21, pp. 118-122, 1976.

2. Decoster, M., Van Cauwenberghe, A.R.: "A Comparative Study of Different Reduction Methods", Journal A, Vol. 17, pp. 68-74, 1976.

3. Binder, Z., Commault, C.: "Une Revue des Méthods de Simplification de Modèles Dinamiques Linéaires Invariants", RAIRO (Automatique), Vol. 12, pp. 199-219, 1978.

4. Wilson, D.A.: "Optimum Solution of Model-Reduction Problem", Proc. IEE, Vol. 117, pp. 1161-1165, 1970.

5. Aplevich, J.D.: "Approximation of Discrete Linear Systems", Int. J. Contr., Vol. 17, pp. 565-575, 1973.

6. Galiana, F.D.: "On the Approximation of Multiple Input-Multiple Output Constant Linear Systems", Int. J. Contr., Vol. 17, pp. 1313-1324, 1973.

7. Aplevich, J.D.: "Gradient Methods for Optimal Linear System Reduction", Int. J. Contr., Vol. 18, pp. 767-772, 1973.

8. Wilson, D.A.: "Model Reduction for Multivariable Systems", Int. J. Contr., Vol. 20, pp. 57-64, 1974.

9. Hirzinger, G., Kreisselmeier, G.: "On Optimal Approximation of High-Order Linear Systems by Low-Order Models", Int. J. Contr., Vol. 22, pp. 399-408, 1975.

10. Decoster, M., Noldus, E., Van Cauwenberghe, A.: "Reduction des Systèms Linéaires Stationaires et Continuus", RAIRO (Automatique), Vol. 10, pp. 47-66, 1976.

11. Obinata, G., Inooka, H.: "A Method for Modeling Linear Time-Invariant Systems by Linear Systems of Low-Order", IEEE Trans. Automat. Contr., Vol. AC-21, pp. 602-603, 1976.

12. Obinata, G., Inooka, H.: "A Method for Reducing the Order of Multivariable Stochastic Systems", IEEE Trans. Automat. Contr., Vol. AC-22, pp. 676-677, 1977.

13. Wilson, D.A., Mishra, R.N.: "Optimal Reduction of Linear Multivariable Systems", Int. J. Contr., Vol. 29, pp. 267-278, 1979.

14. Guardabassi, G.: "Large-Scale Problem Analysis and Decomposition Theory", J. of the Franklin Inst., Vol. 306, pp. 41-62, 1978.

15. Guardabassi, G.: "The Feedback Information Tearing Principle: A Link Between Feedback and Large-Scale Systems Theory", Proc. 20th JACC, pp. 96-100, 1979.

16. Kwakernaak, H., Sivan, R.: "Linear Optimal Control Systems", J. Wiley, New York, 1972.

17. Gantmacher, F.R.: "The Theory of Matrices", Chelsea, New York, 1960.

18. Bartels, R.M., Stewart, G.W.: "Solution of the Matrix Equation AX+XB=C", Comm. ACM, Vol. 15, pp. 820-826, 1972.

19. Kleinman, D.L., Krishna Rao, P.: "Extensions to the Bartels-Stewart Algorithm for Linear Matrix Equations", IEEE Trans. Automat. Contr., Vol. AC-23, pp. 85-87, 1978.

INFORMATION MEASURES IN DECENTRALIZED SYSTEMS

MOHAMED L. EL-SAYED* P.S. KRISHNAPRASAD*
Case Western Reserve University, Cleveland, Ohio, 44106

ABSTRACT

The Kullback-Liebler information measure is used to design a decentralized filtering scheme for a discrete time linear decentralized system corrupted by white Gaussian noise, when the local observation matrices are subject to choice from a finite discrete set.The filtering scheme includes a coordinator with a possibly time varying gain which is optimized using the information measure.

1. INTRODUCTION

Information measures have long been used in statistics to estimate the informational value of some observation sequence. Fisher Information Matrix, Bhattacharya distance and Shannon's conditional entropy are among the most commonly used. In the past few years these measures have been used by some authors in the context of system identification, hypothesis testing and signal detection [1]-[2]. Recently Kullback-Liebler (K-L) information measure [3] has proved to be useful in the analysis of parameter estimation [4], model selection [5] and system identification problems [6].

The problem of designing the observation scheme of a linear system has been considered in many papers. If the sensors in the system can be placed arbitrarily, i.e. the m×n observation matrix is assumed to belong to $R^{m\times n}$, then gradient techniques can be used to find such optimal (reference) matrix. If the sensors can only be placed in a finite number of positions, then the best location would be the one nearest to the reference - possibly non-feasible - observation matrix. For deterministic output feedback systems a quadratic performance function could be used for that purpose. The problem at hand is an estimation problem thus suggesting the use of a statistical measure such as the K-L measure.

The observation design problem is even more important for decentralized systems in which communication links between the controlling stations are considered for further improvement in the system's performance. Since the addition of a number of links can be characterized by augmenting some of the observation matrices, a natural question arises: Which links (or equivalently which observation matrices) to choose from a finite number of possibilities?) Aside from the combinatorial problem, a selection criteria is needed.

In this paper, we consider using K-L measure in the problem of designing a filtering scheme for a discrete time linear decentralized system corrupted by stationary white Gaussian noise, when the observation matrices are subject to choice from a finite discrete set. Because of the nature of the system, the estimation scheme is assumed to consist of local Kalman filters, one for each subsystem, to estimate the subsystem's state and a higher level coordinator. The function of the coordinator is to obtain a global estimate of the overall state of the system using linear combinations of the local estimates and the information about the system's dynamics.

---

*Research supported by the Industrial Sponsors Program, Systems Engineering Department, Case Western Reserve University.

In section 2, we obtain the reference matrices and the K-L distance, and in section 3 we study the observations selection problem and derive the necessary conditions for the optimality of the coordinator's gain. Finally, in section 4 we give some remarks.

2. REFERENCE MATRICES AND KULLBACK-LIEBLER DISTANCE

The problem we treat here is selecting an N-tuple (C_i, i=1,2,...N), where N is the number of subsystems, from a finite number of alternatives. The best candidate is the one which is nearest to some reference tuple (C_i^*, i=1,2,...N) in the K-L distance. So we first derive the equations for the reference tuple and then obtain the expression for the K-L distance.

a. Reference Matrices

The matrices (C_i^*, i=1,2,...N) are chosen such that a function of the steady state error covariance matrix, as resulting from a global Kalman filter, is minimized, i.e. the problem is to minimize trP say, where

$$P=\lim_{t \to \infty} E\{(x_t-\hat{x}_t)(x_t-\hat{x}_t)'\}$$

The subsystems dynamics are given by

$$x^i_{t+1} = \sum_{j=1}^{N} A_{ij}\, x^j_t + v^i_t \qquad (1)$$

$$y^i_t = C_i x^i_t + w^i_t \qquad (2)$$

where x^i_t and y^i_t are the state and output of subsystem i, respectively, and

$$\hat{x}_t = E\,\{x_t/y_{t-1},\ldots y_0\}$$

$$X'_t = [\, x^{1'}_t\, x^{2'}_t \ldots x^{N'}_t \,]$$

The zero mean white Gaussian sequences v^i_t and w^i_t satisfy the usual assumptions. The vectors x^i_t and y^i_t are elements of R^{n_i} and R^{m_i} respectively where

$$\sum_{i=1}^{N} n^i=n \text{ and } \sum_{i=1}^{N} m_i=m$$

In [7], Mehra shows that minimizing trP is equivalent to minimizing $\mathrm{tr}M^{-1}$ or equivalently maximizing

$$m(C_i, \forall i)= (\mathrm{tr}M^{-1})^{-1}$$

where M is the system's observation matrix,

$$M= QQ' \qquad (3)$$

$$Q= [C' \mid A'C' \mid A^{2'}C' \mid \ldots \mid A^{n-1}C'] \qquad (4)$$

C= block diag (C_i)

and $A=\{A_{ij}\}$

For the problem to be well posed, we introduce an upper bound on the energy of the sensors (or signal to noise ratio), say

$$\|C\| \leq 1 \qquad (5)$$

Since $m(\alpha c)= \alpha^2 m(c)$, where is any nonzero scalar, then $m(\alpha c)$ can be made arbitrarily large by increasing α. Thus constraint (5) is binding at the optimal solution, i.e.

$$\mathrm{tr}CC'= \sum_{i=1}^{N} \mathrm{tr}C_iC'_i = 1 \qquad (5')$$

We form the Lagrangian

$$L= m(C_i, \forall_i) - \rho(\sum_{i=1}^{N} trC_iC_i' -1)$$

then

$$\partial L/\partial C_i' = 0 \rightarrow \partial m/\partial C_i' = 2\rho C_i' \qquad i=1,2,\ldots,N \tag{6}$$

From (6), it is easy to show that

$$\sum_{i=1}^{N} \frac{1}{2} tr[(\partial m/\partial C_i')'C_i'] = \frac{1}{2} tr \sum_{i=1}^{N} (2\rho C_i')'C_i' = \rho$$

Also using Euler's Therom ([8], eqt.(45))

$$\frac{1}{2} tr \sum_{i=1}^{N} [(\partial m/\partial C_i')'C_i'] = \frac{1}{2} tr [(\partial m/\partial C')'C'] = m$$

i.e. $m = \rho$

Thus (6) reduces to

$$\partial m/\partial C_i' = 2mC_i' \qquad i=1,2,\ldots,N \tag{6'}$$

Now

$$\partial m/\partial C_i' = -m^2\partial[tr(QQ')^{-1}]/\partial C_i'$$

$$= -m^2\partial[(vec\ I_n)'(I_n \otimes (QQ')^{-1})(vec\ I_n)]/\partial C_i' \tag{7}$$

where I_n is the n×n identity matrix, vec I_n is I_n arranged column-wise as a column vector and $\otimes$ denotes Kronecker product. After some calculations, (7) reduces to

$$\begin{aligned}\partial m/\partial C_i' = m^2 [I_{n_i} \otimes (vec\ I_n)' U_{n\times n}]\{[(I_{n_i} \otimes (QQ')^{-1}\bar{Q}\ U_{n\times n}) \otimes (\partial C/\partial C_i \otimes \\ I_n)'(I_{m_i} \otimes U_{m\times n}\ Q'(QQ')^{-1}) + (I_{n_i} \otimes (QQ')^{-1}Q\ U'_{m\times n}) \\ (\partial C/\partial C_i' \otimes I_n)(I_{m_i} \otimes U'_{n\times n}\bar{Q}^{-1}(QQ')^{-1})]\otimes I_n\}[I_{m_i} \otimes U_{n\times n}(vec\ I_n)]\end{aligned}$$

where

$$\bar{Q} = [I \mid A' \mid A'^2 \mid \ldots. \mid A'^{n-1}]$$

$$\bar{U}_{m\times n} = \sum_{j}^{m} \sum_{k}^{n} E_{jk}^{(m\times n)} \otimes E_{ki}^{(n\times m)}$$

If $n_1=n_2= \cdots =n_N$ and $m_1=m_2= \cdots =m_N$, then $\partial C/\partial C_i$ could be written as

$$\partial C/\partial C_i = \partial(\sum_{i=1}^{N} E_{ii}^{(m\ n)} \otimes C_i)/\partial C_i$$

$$= (I_{m_i} \otimes U_{m\times m_i})(\bar{U}_{m_i\times n_i} \otimes E_{ii}^{(m\ n)})(I_{n_i} \otimes U_{n_i\times n}) \tag{9}$$

Similarly

$$\partial C/\partial C_i' = (I_{n_i} \otimes U_{m\times m_i})(U_{m_i\times n_i} \otimes E_{ii}^{(m\times n)})(I_{m_i} \otimes U_{n_i\times n}) \tag{10}$$

From (5'), (6'), and (8)-(10) $(C_i^*, \forall i)$ could be obtained .

b. Kullback-Liebler Information Measure

The K-L distance is defined as

$$I_t(*,s) = E_* \log [\, f_*(x_t/y_{t-1},\ldots,y_o)/f_s(x_t/y_{t-1},\ldots,y_o)]$$

where $f_*(.)$ is the conditional density of the observations when a Kalman filter is applied to the overall system (1)-(2) while the reference tuple $(C_i^* \,,\ i=1,2,\ldots,N)$ obtained in (a) is used. $f_s(.)$ is the conditional density of the observations for the actual system using local Kalman filters, the coordinator and the N-tuple $(C_i^s \,,\ i=1,2,\ldots,N)$. The expectation is taken with respect to $f_*(.)$.

Before we derive the equations for the K-L measure , we give first expressions for the local Kalman filters in the steady state , namely

$$\bar{x}^i_{s,t} = A_{ii}\bar{x}^i_{s,t-1} + K^i(y^i_{t-1} - C^s_i\bar{x}^i_{s,t-1}) \qquad i=1,2,\ldots,n \tag{11}$$

where

$\bar{x}^i_{s,t}$ = estimated state of subsystems i at time t with actual observation matrix C^s_i

K^i = gain of the Kalman filter for subsystem i

Equations (11) could be rewritten as

$$\bar{x}_{s,t} = A_d\bar{x}_{s,t-1} + \bar{K}\,(y_{t-1} - C^s\bar{x}_{s,t-1}) \tag{12}$$

where

A_d = block diag(A_{ii})

K = block diag(K^i) and

$\bar{x}$ = col$(\bar{x}^i_{s,t})$

Since the output of the coordinator will be

$$x_{s,t} = G\,\bar{x}_{s,t}$$

where G is the coordinator's gain , then x_t will be unbiased if

$E\,\{x_o\} = x_{s,o}$, and

$$AG - G\bar{K}C^sG - G(A_d - \bar{K}C^s) = 0 \tag{13}$$

The error covariance matrix defined by

$M^s_t = E_s\{(x_t - x_{s,t})(x_t - x_{s,t})\}$ will satisfy the linear matrix equation

$$M^s_t = (A - G\bar{K}C^s)M_{s,t-1}(A - GK\bar{C}^s)' + V + G\bar{K}W^s\bar{K}'G' \tag{14}$$

where

V = covariance of $v_t = \text{col}(v^i_t)$

W^s = covariance of $w^s_t = \text{col}(w^{s,i}_t)$

If we let

$z'_t = [x'_t \,\vdots\, \bar{x}'_{s,t}]$ and $\eta'_t = [v'_t \,\vdots\, w^{s\,\prime}_t]$

then following the approach in [5] , it can be shown that

$$z_{t+1} = F_s z_t + D_s \eta_t \qquad (15)$$

where

$$F_s = \left[\begin{array}{c|c} A & 0 \\ \hline \bar{K}C^* & \bar{A}-\bar{K}C^s \end{array}\right] \quad \text{and} \quad D_s = \left[\begin{array}{c|c} I & 0 \\ \hline 0 & \bar{K} \end{array}\right]$$

Now let

$$\psi^*_{s,t} = E_*\{z_t z'_t\}$$

$$\Gamma^*_{s,t} = E_*\{(x_t - x_{s,t})(x_t - x_{s,t})'\}$$

then

$$\psi^*_{s,t+1} = F_s \psi^*_{s,t} F'_s + D_s Q_s D'_s \qquad (16)$$

and

$$\Gamma^*_{s,t} = T \Gamma^*_{s,t} T' \qquad (17)$$

where

$$Q = \left[\begin{array}{c|c} V & 0 \\ \hline 0 & W^s \end{array}\right] \quad \text{and} \quad T = [\, I \mid -G \,]$$

The steady state solution of (16), and hence (17), exists if A is stable and (A,C^s) is observable.

Next define M^*_t as the state estimation error covariance and $\hat{x}_{*,t}$ as the state estimate, corresponding to the reference model i.e.

$$M^*_t = E_*\{(x_t - \hat{x}_{*,t})(x_t - \hat{x}_{*,t})'\}$$

The conditional probability densities of x_t given the past data $(y_{t-1}, y_{t-2}, \ldots, y_o)$ are

$$f_s(x_t/y_{t-1}, \ldots, y_o) = [(2\pi)^n \,|M^s|\,]^{-1/2} \exp\{-1/2(x_t - \hat{x}_{s,t})' M^{s^{-1}} (x_t - \hat{x}_{s,t})\}$$

and

$$f_*(x_t/y_{t-1}, \ldots, y_o) = [(2\pi)^n \,|M^*|\,]^{-1/2} \exp\{-1/2(x_t - \hat{x}_{*,t})' M^{*^{-1}} (x_t - \hat{x}_{*,t})\}$$

Thus the K-L distance is given by

$$I(*,s) = 1/2 \log [\,|M^s| / |M^*|\,] + 1/2 \operatorname{tr} (M^{s^{-1}} \Gamma^*_s) - n/2 \qquad (18)$$

where M^s, M^*, Γ^*_s and $I(*,s)$ are the steady state values of M^s_t, M^*_t, $\Gamma^*_{s,t}$ and $I_t(*,s)$ respectively.

The above equations are summarized in the block diagram of figure (1). Notice that even if $C^*=C^s$ and $W^*=W^s$ then $I(*,s)$ in (18) will not be zero since $M^*=M^s$ because of the decentralization of the filters. We also notice that (18) is an implicit function of G and $(C^*_i,\ i=1,2,\ldots,N)$, thus we have a combined optimization problem which we study next.

3. OPTIMIZATION PROBLEM

a. Coordinator's Gain

For a certain N-tuple , say $(C_i^k , i=1,2,\ldots,N)$, the coordinator's gain is optimized by minimizing (18) subject to all constraints. So the Lagrangian

$$H = I(*,k) + \mathrm{tr}[M^k - (A-G\bar{K}C^k)M^k(A-G\bar{K}C^k)' - V - G\bar{K}W^k\bar{K}'G']\mu'$$
$$+ \mathrm{tr}[\Gamma_k^* - T\psi_k^* T']\phi' + \mathrm{tr}[AG - G(A_d - \bar{K}C^k) - G\bar{K}C^kG]\gamma' \qquad (19)$$

is minimized with respect to G , M^k , μ , ϕ , γ , and Γ_k^*.

The necessary conditions for optimality are

$$\partial H/\partial \Gamma_k^* = 0 \rightarrow M^{k^{-1}} = -2\,\phi \qquad (20)$$

$$\partial H/\partial M_k = 0 \rightarrow M^k\mu M^k - M^k(A - G\bar{K}C^k)'\mu(A - G\bar{K}C^k)M^k + 1/2\, M^k$$
$$-1/2\, \Gamma_k^* = 0 \qquad (21)$$

$$\partial H/\partial G = 0 \rightarrow [2AM^kC^{k\prime}\bar{K}' - (G + G')\bar{K}C^kM^kC^{k\prime}\bar{K}' - (G + G')\bar{K}W^k\bar{K}']$$
$$+ A'\gamma - \gamma[A_d - \bar{K}(C^k + G)]' - \gamma'G\bar{K}$$
$$+S \sum_{i=1}^{2} (e_j \otimes I_{2n})\psi_k^* E_j = 0 \qquad (22)$$

where

$E_1' = [\, I_n' \mid O_n' \,]$, $E_2 = [\, O_n' \mid I_n' \,]$, $e_1 = \begin{bmatrix}1\\0\end{bmatrix}$, $e_2 = \begin{bmatrix}0\\1\end{bmatrix}$ and 0_n is the $n \times n$ zero matrix and

$$S = \phi[O_n \mid I_n \mid I_n \mid -(G + G')] \qquad (23)$$

The remaining conditions

$\partial H/\partial\gamma = \partial H/\partial\mu = \partial H/\partial\phi = 0$

are given by (13) , (14) , and (17) respectively .

b. Observation Matrices

For a given coordinator gain the best set of observation matrices $(C_i^k , i=1,2,\ldots,N)$ is the one which minimizes (18) . Since the number of alternatives is finite , a simple search procedure could be used .

Since both G and $(C_i^k , i=1,2,\ldots,N)$ have to be optimized simultaneously , an iterative procedure is suggested . We first start with an initial guess for G which satisfies (13) and obtain the best N-tuple $(C_i^j , i=1,2,\ldots,N)$, $j\epsilon\{1,2,\ldots,\Omega\}$ where Ω is the number of alternatives . An optimal value for G is calculated using the conditions derived in section 3.a and the process is repeated untill convergence is achieved. The solution obtained is a "person-to-person" optimal but may not be globally optimal.

4. CONCLUSION

We conclude with some remarks:

1. Equations (5') and (6') are only necessary for local optimality since the optimization problem is nonconvex . On the other hand , the optimal M is unique because of the strict convexity of $\mathrm{tr}(M)^{-1}$ in M.
2. The above analysis can be easily extended to the case of a dynamic and/or time varying coordinator for the finite horizon problem . An affine coordinator law could also be used thus increasing the degrees of freedom of the problem.
3. Although the optimal coordinator gain - $\tilde{G}$ - may be singular , this does not imply any loss of information about the system , unless one column of $\tilde{G}$ is identically zero.
4. There are still some open questions relating to the existence of non-trivial solutions to (13) and to the optimality conditions in section (2.a) and (3.a) . We expect further work to be done in that respect in the near future .

REFERENCES

[1] R.G. Gallager , Information Theory and Reliable Communication , John Wiley , 1968.

[2] S. Arimoto , "Information-Theoretical Considerations on Estimation Problems ", Information and Control , 19 , pp. 181-194 , 1971 .

[3] S. Kullback , Information Theory and Statistics , New York : Wiley 1959.

[4] R.M. Hawkes and J.B. Moore , " Performance Bounds for Adaptive Estimation ", Proc. IEEE , Aug. 1976.

[5] Y.Baram and N. Sandell , " An Information Theoretic Approach to Dynamical Systems Modeling and Identification ", IEEE Trans. Aut. Control , vol. AC-23 ,no 1 , pp.61-66 , Feb. 1978.

[6] H. Akaike , " A New look at the Statistical Model Identification ", IEEE Trans. Aut. Control , vol. AC-19 , pp.667-674 , Dec. 1974.

[7] R. Mehra , " Optimization of Measurement Schedules and Sensor Designs for Linear Dynamical Systems " , IEEE Trans. Aut. Control , vol. AC-21 , no1 , pp.55-64 ,Feb. 1976.

[8] P.C. Muller and H.I. Weber , " Analysis and Optimization of Certain Qualities of Controllability and Observability for Linear Dynamical Systems " , Automatica , vol.8 ,pp. 237-246 , 1972.

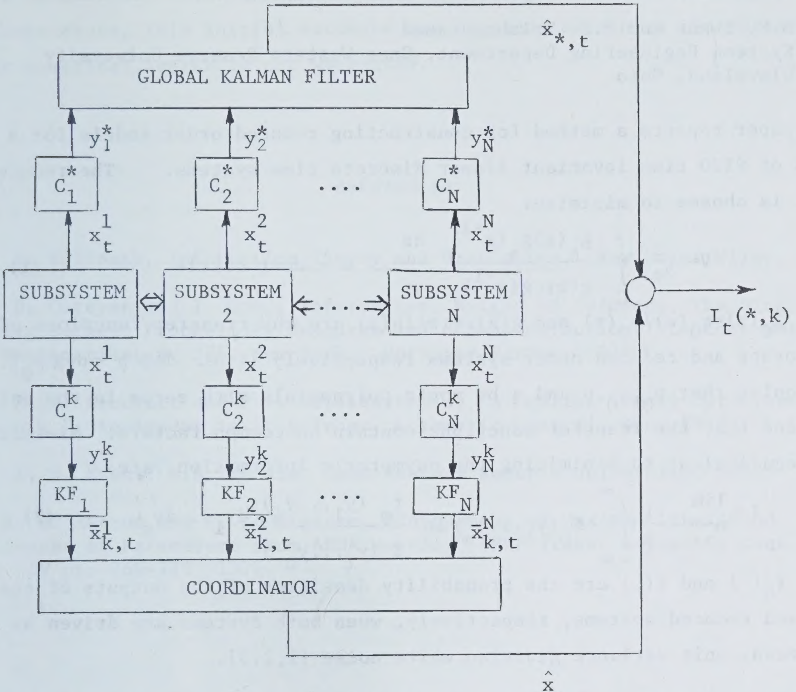

GLOBAL KALMAN FILTER
$\hat{x}_{*,t}$
y_1^*
y_2^*
y_N^*
C_1^*
C_2^*
....
C_N^*
x_t^1
x_t^2
x_t^N
SUBSYSTEM 1
SUBSYSTEM 2
SUBSYSTEM N
$I_t(*,k)$
x_t^1
x_t^2
x_t^N
C_1^k
C_2^k
....
C_N^k
y_1^k
y_2^k
y_N^k
KF_1
KF_2
....
KF_N
$\bar{x}_{k,t}^1$
$\bar{x}_{k,t}^2$
$\bar{x}_{k,t}^N$
COORDINATOR
$\hat{x}_{s,t}$

Figure 1

MODEL REDUCTION USING AN INFORMATION RATE CRITERION

B.S. Evans and P.S. Krishnaprasad
Systems Engineering Department, Case Western Reserve University
Cleveland, Ohio

This paper reports a method for constructing reduced order models for a class of SISO time invariant linear discrete time systems. The reduced model is chosen to minimize:

$$V = \frac{1}{2\pi i} \oint_C \frac{g_0(z) g_0(z^{-1})}{g(z)\; g(z^{-1})} \frac{dz}{z} \qquad C = \text{unit circle} \tag{1}$$

where $g_0(z)=q_0(z)/p_0(z)$ and $g(z)=q(z)/p(z)$ are the transfer functions of the full order and reduced order systems respectively (i. e. $\deg p < \deg p_0$). We require that p_0, q_0, p and q be monic polynomials with zeros in the unit disc and that the transfer functions contain no common factors. Minimizing V is equilvalent to minimizing the asymptotic information rate:

$$I = \lim_{n\to\infty} \frac{1}{n} \int_{-\infty}^{\infty} f_0(y_1 \ldots y_n) \ln \frac{f_0(y_1 \ldots y_n)}{f(y_1 \ldots y_n)} dy_1 \ldots dy_n \tag{2}$$

where $f_0(.)$ and $f(.)$ are the probability densities of the outputs of the full and reduced systems, respectively, when both systems are driven by zero mean, unit variance gaussian white noise [1,2,3].

Let p and q denote the coefficients of p(z) and q(z). Then the reduced systems must satisfy:

$$\text{(a)} \quad \frac{\partial V}{\partial q} = 0 \qquad\qquad \text{(b)} \quad \frac{\partial V}{\partial p} = 0 \tag{3}$$

It can be shown that for any fixed q, (3b) has a unique solution which is a relative minimum. Furthermore, this solution may be found exactly be a series of recursions on the reduced model order [4]. Thus, one need only be concerned with finding the zeros of the reduced model.

It is generally not possible to solve (3a) analytically, and there may be multiple solutions, some of which may be local maxima or saddle points [5]. Thus, when descent methods are used to solve (4a) the choice of the initial estimate of the solution is crucial to obtaining a global minimum. Based on heuristic considerations and numerical experimentation, a good choice appears to be to choose q so as to minimize:

$$\frac{1}{2\pi i} \oint_C \frac{q(z) q(z^{-1})}{q_0(z) q_0(z^{-1})} \frac{dz}{z} \tag{4}$$

This minimization can be solved by the same method used for solving (4b). In some cases, this initial estimate may be close enough to the optimum that numerical search is not warrented.

References

1. S. Kullback, Information Theory and Statistics. New York: Wiley, 1951.

2. D. Osteyee and I. Good, Information, Weight of Evidence, The Singularity Between Probability Measures and Signal Detection. Lecture Notes in Mathematics, n. 376. New York: Springer-Verlag, 1974.

3. R. W. Brockett and P.S. Krishnaprasad, "A Scaling Theory for Linear Systems," to appear in IEEE Trans. Automatic Control, Dec. 1979.

4. B. W. Evans, M.S. Thesis, Case Western Reserve University, 1979.

5. K. J. Astrom and T. Soderstrom, "Uniqueness of Maximum Likelihood Estimates of Parameters from ARMA Models," IEEE Trans. Automatic Control. Vol. 19 pp. 769-773, 1976.

QUADRIPHASE DIRECT-SEQUENCE SPREAD-SPECTRUM MULTIPLE-ACCESS COMMUNICATION VIA FADING CHANNELS

DAVID E. BORTH
School of Electrical Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332

ABSTRACT

The effect of Rician or specular-plus-diffuse fading on the performance of quadriphase spread-spectrum multiple-access communications is analyzed. Two classes of wide-sense-stationary uncorrelated-scattering fading channels are considered: time-selective fading channels and frequency-selective fading channels. Utilizing a "generalized" quadriphase spread-spectrum multiple-access system model, an expression is obtained for the average signal-to-noise ratio at the output of a correlation receiver in terms of the channel covariance function for the fading process, the complex signature sequences, and the thermal noise density of the additive channel noise. For quadriphase systems employing quadriphase-coded carriers and for specific channel covariance functions, the signal-to-noise ratio becomes a function of the complex discrete aperiodic autocorrelation functions of the signature sequences. A comparison of the performance of quadriphase and biphase spread-spectrum multiple-access communication systems operating over fading channels is made.

INTRODUCTION

In recent years a multiple-access communication technique known as spread-spectrum multiple-access (SSMA) has generated much interest in the engineering community. The reason for this interest in the spread-spectrum technique is that in addition to the multiple-access capabilities that can be provided by the technique, other desirable qualities are simultaneously provided including increased immunity from interference and jamming, compatibility with other (non-spread-spectrum) systems operating in the same frequency band, and with the proper choice of modulation method, increased immunity from the effects of multipath distortion. Two major types of SSMA are frequency-hopped SSMA (FH/SSMA) and direct sequence SSMA (DS/SSMA), also often referred to as pseudo-noise or phase-coded SSMA. For the remainder of this paper we shall be concerned only with a particular type of DS/SSMA known as quadriphase direct-sequence spread-spectrum multiple-access (QDS/SSMA).

The motivation for studying the performance of QDS/SSMA systems stems primarily from the following two considerations: (1) for a given bandwidth and data rate, QDS/SSMA systems can employ signature sequences which are

---

This research was supported in part by the Joint Services Electronics Program at the University of Illinois, Urbana, Illinois 61801, under Contract N00014-79-C-0424.

twice the length of the binary sequences in a biphase DS/SSMA system and (2) the generalization to quadriphase codes as signature sequences allows the consideration of larger classes of signature sequences for use in DS/SSMA systems. The second consideration is perhaps of stronger interest to the SSMA system designer since it offers promise of obtaining improved performance over biphase DS/SSMA systems due to the larger selection of signature sequences available. In a QDS/SSMA communications system with K users employing an additive white Gaussian noise (AWGN) channel, the output of each of the synchronized correlation receivers consists of a desired signal component, an interference component due to the quadrature component of the quadriphase signal, K-2 interfering signals due to the remaining users, and an additive channel noise component. The signal-to-noise ratio (SNR) at the output of the correlation receiver has been previously analyzed in [5,6] for an AWGN channel. This signal-to-noise ratio has been shown in [9] to be an accurate measure of the average bit error probability (P_e) for biphase DS/SSMA systems of interest. In the definition of signal-to-noise ratio, the noise is taken to be the interference due to the quadrature component of the signal, the K-2 interfering signal components, plus the channel noise component. It has been shown [5,6] that the signal-to-noise ratio is completely specified by the complex aperiodic autocorrelation functions for the K complex signature sequences and the channel noise spectral density.

In the present paper we consider QDS/SSMA communication via fading channels. The type of fading considered is Rician fading; that is, for a single transmitted signal, the received signal consists of a strong specular signal (the desired signal) plus a weaker diffuse or Rayleigh-fading component. For fading that is not Rician (i.e., no strong specular component), the direct-sequence form of spread-spectrum communication would not generally be useful, since it could not be coherently demodulated. The fading process is modeled as a unit-energy complex Gaussian process with a covariance function corresponding to that of a wide-sense-stationary uncorrelated-scattering (WSSUS) fading channel, a channel general enough to exhibit both time- and frequency-selective fading.

SYSTEM MODEL AND ANALYSIS

System Model

The QDS/SSMA system model that we consider is taken from [6] and is shown in Figure 1. For each k ($1 \le k \le K$) the k-th user's data signal $b_k(t)$ is a sequence of statistically independent, unit amplitude, positive and negative, rectangular pulses of duration T_d. Each user is assigned a periodic code waveform $a_k(t)$ consisting of a sequence of rectangular pulses of duration T_c. The code waveform for the in-phase (I) branch of the k-th transmitter is therefore given by

$$\alpha_{2k}(t) = \sum_{\ell=-\infty}^{\infty} \alpha_{\ell}^{(2k)} p_{T_c}(t - \ell T_c) \tag{1a}$$

while the code waveform for the quadrature (Q) branch of the k-th transmitter is given by

$$\alpha_{2k-1}(t) = \sum_{\ell=-\infty}^{\infty} \alpha_{\ell}^{(2k-1)} p_{T_c}(t - \ell T_c), \tag{1b}$$

where $\{\alpha^{(2k)}\}$ and $\{\alpha^{(2k-1)}\}$ are the signature sequences associated with the I and Q branches of the k-th transmitter and take on values from the set $\{0, \frac{\pi}{2}, \pi, {}^{3\pi}/2\}$ and $p_\tau(t) = 1$ for $0 \le t < \tau$, $p_\tau = 0$ otherwise. We assume that $T_d/T_c = N$ where N is the period of the signature sequence for each user. The two signals $b_k(t)$ and $\alpha_k(t)$ are used to simultaneously phase modulate a cosinusoidal carrier signal having power P at a frequency ω_c with phase θ_k. In the absence of fading, the received signal at the input to a receiver is given by

$$r(t) = n(t) + \sum_{k=1}^{K} s_k(t - \tau_k) \tag{2}$$

where

$$\begin{aligned} s_k(t) &\triangleq s_k^{(I)}(t) + s_k^{(Q)}(t) \\ &= \sqrt{P}\,\mathrm{Re}[\tilde{\alpha}_{2k}(t)\, b_{2k}(t) \exp[j(\omega_c t + \theta_k)]] \\ &\quad + \sqrt{P}\,\mathrm{Re}[\tilde{\alpha}_{2k-1}(t) b_{2k-1}(t) \exp[j(\omega_c t + \theta_k + \pi/2)]], \end{aligned} \tag{3}$$

$n(t)$ is AWGN with spectral density $N_0/2$ and τ_k is due to the propagation time delay and time asynchronism for the k-th user. In writing (3) we have used the complex forms of the code waveforms $\alpha_{2k}(t)$ and $\alpha_{2k-1}(t)$ defined by

$$\tilde{\alpha}_{2k}(t) = e^{j\alpha_{2k}(t)} \tag{4a}$$

and

$$\tilde{\alpha}_{2k-1}(t) = e^{j\alpha_{2k-1}(t)} \tag{4b}$$

respectively. It should be noted that the QDS/SSMA system model described above includes as special cases, QDS/SSMA systems employing either orthogonal biphase-coded carriers (i.e., $\alpha_{2k}(t)\epsilon\{0,\pi\}$, $\alpha_{2k-1}(t)\epsilon\{0,\pi\}$, $1\leq k\leq K$) or orthogonal quadriphase coded carriers (i.e., $\alpha_{2k}(t) = \alpha_{2k-1}(t)\epsilon\{0,\frac{\pi}{2},\pi,\frac{3\pi}{2}\}$).

Channel Model

For short time durations on the order of the bit duration T_d, it has been shown [1] that the WSSUS channel model is a suitable model for practical radio channels. Furthermore, as noted above, the WSSUS model is of interest because it is the simplest channel model of physical interest exhibiting both time- and frequency-selective fading behavior. Hence, in its most general form, the WSSUS channel is called a doubly-spread fading channel. In the course of this paper, we shall focus attention on the performance of QDS/SSMA communication via a general WSSUS channel which has a strong specular component (i.e., Rician fading). Thus if $\mathrm{Re}\{x_k(t)\exp[j\omega_c t + j\theta_k]\}$ is the input to the channel from the kth transmitter, then the corresponding output $y_k(t)$ is given by

$$y_k(t) = \mathrm{Re}\{u_k(t-\tau_k)\exp[j\omega_c(t-\tau_k) + j\theta_k]\} + n(t) \ , \tag{5}$$

where

$$u_k(t) \triangleq \gamma_k \int_{-\infty}^{\infty} \beta_k(\tau,t)x_k(t-\tau)d\tau + x_k(t) \ . \tag{6}$$

The non-negative real parameter γ_k is the transmission coefficient for the fading channel as seen by the k-th signal and $\beta_k(\tau,t)$ is a zero-mean unit-energy complex Gaussian random process which will be referred to as the fading process. The covariance function for the fading process is

$$\Lambda_k(\tau,\sigma,t,s) \triangleq \frac{1}{2} E\{\beta_k(\tau,t)\beta_k^*(\sigma,s)\}$$

$$= \rho_k(\tau,t-s)\delta(\tau-\sigma) \ . \tag{7}$$

Analysis

From (5) we see that for the QDS/SSMA system considered above, $x_k(t) = \sqrt{P}\,[\tilde{\alpha}_{2k}(t)b_{2k}(t) + \tilde{\alpha}_{2k-1}(t)b_{2k-1}(t)\,e^{j\pi/2}]$ and from (2) and (3) we obtain

$$r(t) = n(t) + \sum_{k=1}^{K} \mathrm{Re}[u_k(t - \tau_k)\exp\{j(\omega_c t + \phi_k)\}] \tag{8}$$

where $\phi_k \triangleq \theta_k - \omega_c \tau_k$. If the received signal $r(t)$ is the input to a correlation receiver matched to $s_i^{(I)}(t)$, the corresponding output is

$$Z_i^{(I)} = \int_0^{T_d} r(t)\mathrm{Re}\{\tilde{\alpha}_{2i}(t - \tau_i)\exp(j\omega_c t + j\phi_i)\ dt . \tag{9}$$

Without loss of generality we shall assume $\tau_i = 0$ and $\theta_i = 0$. Substituting (8) into (9) and neglecting the double-frequency terms, we obtain

$$Z_i^{(I)} = \int_0^{T_d} n(t)\cos(\omega_c t + \alpha_{2i}(t))dt + \sum_{k=1}^{K} \frac{1}{2} \int_0^{T_d} \mathrm{Re} \left\{u_k(t - \tau_k)e^{j\phi_k}\tilde{\alpha}_{2i}^{*}\right\}dt . \tag{10a}$$

In a similar manner it may be shown that if $r(t)$ is the input to a receiver matched to $s_i^{(Q)}(t)$, the corresponding output is

$$Z_i^{(Q)} = -\int_0^{T_d} n(t)\sin(\omega_c t + \tilde{\alpha}_{2i-1}(t))dt + \sum_{k=1}^{K} \frac{1}{2} \int_0^{T_d} \mathrm{Re} \left\{u_k(t - \tau_k)e^{j\phi_k}\tilde{\alpha}_{2i-1}^{*} e^{-j\pi/2}\right\}dt . \tag{10b}$$

As noted in the introduction, two measures of performance of a DS/SSMA over an AWGN channel that have been considered in the published literature are SNR_i and P_e, which for many cases of interest, is simply related to SNR_i by $P_e \approx 1 - \Phi(\mathrm{SNR}_i)$. We conjecture that a similar relationship between SNR_i and P_e also holds for QDS/SSMA communication via fading channels.

As in [5], the phase angles, time delays, and data symbols for the k-th signal ($k \neq i$) are modeled as mutually independent random variables which are uniformly distributed on $[0,2\pi)$, $[0,T_d)$, and $\{+1, -1\}$, respectively. SNR_i may then be defined in terms of probabilistic averages conditioned upon $b_i(t) = +1$ for $0 \le t \le T_d$.

We may rewrite (10) as $Z_i^{I(Q)} = N_i^{I(Q)} + D_i^{I(Q)} + F_i^{I(Q)} + I_i^{I(Q)}$ where $N_i^{I(Q)}$, $D_i^{I(Q)}$, $F_i^{I(Q)}$, and $I_i^{I(Q)}$ are the AWGN component, the i-th specular signal component, the i-th faded signal component, and the i-th interference signal component, respectively, for the inphase (quadrature) branch of the i-th receiver.

We define the signal-to-noise ratio $\mathrm{SNR}_i^{I(Q)}$ at the output of the in-phase (quadrature) branch of the i-th correlation receiver to be

$$\mathrm{SNR}_i \triangleq E(D_i^{I(Q)})[\mathrm{Var}\ D_i^{I(Q)} + \mathrm{Var}\ F_i^{I(Q)} + \mathrm{Var}\ I_i^{I(Q)} + \mathrm{Var}\ N_i^{I(Q)}]^{-\frac{1}{2}} , \tag{11}$$

which depends on both the channel correlation functions ρ_k $(1 \le k \le K)$ and the signature sequences $\{\alpha_n^{(k)}\}$. In the next two subsections, we will evaluate (11) for two special classes of WSSUS channels.

QDS/SSMA COMMUNICATION VIA TIME-SELECTIVE FADING CHANNELS

Analysis

In this subsection, we consider one subclass of WSSUS channels known either as time-selective fading channels [8] or as channels dispersive only in frequency [4]. The channel correlation function for a time-selective fading channel is [4]

$$\rho_i(\tau, t - s) = \rho_i(0, t - s)\delta(\tau) \triangleq r_i(t - s)\delta(\tau) . \tag{12}$$

Using (12), we find the variance of the diffuse component of the i-th signal at the output of the in-phase branch of the receiver is

$$\mathrm{Var}\ F_i^{(I)} = \gamma_i^2 \frac{P}{2} \int_0^{T_d} r(u)(T - u)du + \gamma_i^2 \frac{P}{4} \cdot$$

$$\mathrm{Re}\left\{ \int_0^{T_d} r_i(u)\hat{\Lambda}_{2i-1,2i}(u)\ du + \int_{-T_d}^{0} r(u)\Lambda_{2i-1,2i}(u)du \right\} \tag{13}$$

where $\hat{\Lambda}_{j,k}(u)$ and $\Lambda_{j,k}(u)$ are defined by

$$\hat{\Lambda}_{j,k}(u) = \int_u^{T_d} \tilde{\alpha}_j(v)\tilde{\alpha}_j^*(v - u)\tilde{\alpha}_k^*(v)\tilde{\alpha}_{2i-1}(v - u)dv \tag{14a}$$

$$\Lambda_{j,k}(u) = \int_0^{T_d - u} \tilde{\alpha}_j(v)\tilde{\alpha}_j^*(v + u)\tilde{\alpha}_k^*(v)\tilde{\alpha}_k(v + u)dv \tag{14b}$$

for $0 \le u \le T_d$. Similarly, using (12), the variance of the interference component I_i at the output of the in-phase branch of the receiver is

$$\mathrm{Var}\ I_i^{(I)} = \frac{P}{4} \sum_{\substack{k-1 \\ k \neq 1}}^{K} \Bigg\{ \gamma_k^2 \frac{2}{T_d} \int_0^{T_d} r_k(v) \left[\mathrm{Re} \left\{ \hat{R}_{2k}^{*}(v) \hat{R}_{2i}(v) + \hat{R}_{2k-1}^{*}(v) \hat{R}_{2i}(v) \right\} \right] dv$$

$$+ \frac{T_c^3}{6T_d} \left[r_{2k-1,2i} + r_{2k,2i} \right] \Bigg\} \tag{15}$$

where $\hat{R}_j(v)$ and $R_j(v)$ are the complex continuous-time partial autocorrelation functions of the code signature sequence $\alpha^{(j)}$ and $r_{j,k}$ is the multiple-access interference parameter as defined in [5]. Finally from [6], the variance of the i-th specular signal component $D_i^{(I)}$ at the output of the in-phase branch of the receiver is

$$\mathrm{Var}\ D_i^{(I)} = \frac{PT_c^2}{4} \left[\mathrm{Im} \left\{ C_{2i-1,2i}(0) \right\} \right]^2 \tag{16}$$

where $C_{x,y}(\ell)$ is the complex discrete-time aperiodic cross-correlation function of sequences x and y. From (11), (13), (15), and (16), $\mathrm{SNR}_i^{(I)}$ for a QDS/SSMA system operating over a time-selective fading channel is found to be

$$\mathrm{SNR}_i^{(I)} = \Bigg\{ \frac{N_0}{2\varepsilon} + \gamma_i^2 \frac{2}{T_d} \int_0^{T_d} r_i(u)(T_d - u)\,du + \gamma_i^2 \frac{1}{T_d^2} \mathrm{Re} \left\{ \int_0^{T_d} r_i(u) \hat{\Lambda}_{2i-1,2i}(u)\,du \right.$$

$$\left. + \int_{-T_d}^{0} r_i(u) \Lambda_{2i-1,2i}(u)\,du \right\} + \left[\mathrm{Im} \left\{ C_{2i-1,2i}(0) \right\} /N \right]^2$$

$$+ \frac{1}{T_d^2} \sum_{\substack{k=1 \\ k \neq 1}}^{K} \Bigg[\gamma_k^2 \frac{2}{T_d} \int_0^{T_d} r_k(v) \mathrm{Re} \left\{ \hat{R}_{2k-1}^{*}(v) \hat{R}_{2i}(v) + \hat{R}_{2k}^{*}(v) \hat{R}_{2i}(v) \right\} \Bigg] dv$$

$$+ \frac{T_c^3}{6T_d} \left[r_{2k-1,2i} + r_{2k,2i} \right] \Bigg] \Bigg\}^{-\frac{1}{2}} \tag{17}$$

where $\varepsilon \triangleq \frac{1}{2} PT_d$ is the average energy per data bit.

If quadriphase coded carriers are employed by the QDS/SSMA system, it may be shown that the expression for $\mathrm{SNR}_i^{(I)}$ may be further simplified to

$$\mathrm{SNR}_i^{(I)} = \Bigg\{ \frac{N_0}{2\varepsilon} + \frac{4\gamma_i^2}{T_d^2} \int_0^{T_d} r_i(u)(T-u)\,du + \frac{1}{T_d^3} \sum_{\substack{k=1 \\ k \neq i}}^{K}$$

$$\left\{4\gamma_k^2 \int_0^{T_d} r_k(v)\ \mathrm{Re}\left\{\hat{R}_{2k}^*(v)\hat{R}_{2i}(v)\right\} dv + \frac{T_c^3}{3}\, r_{2k,2i}\right\}^{-\frac{1}{2}} . \tag{18}$$

Finally, although the results presented above were for the in-phase branch of the receiver, the corresponding results for the quadrature branch of the receiver may be found from the given results by a simple modification of the appropriate subscripts.

QDS/SSMA COMMUNICATIONS VIA FREQUENCY-SELECTIVE FADING CHANNELS

Analysis

The dual of the time-selective fading channel is another subclass of a WSSUS channel known either as a frequency-selective fading channel [8] or a channel dispersive only in time [4]. In this subsection we will present results obtained on the performance of QDS/SSMA communications via frequency-selective fading channels.

The channel correlation function for a frequency-selective fading channel is [4]

$$g_i(\tau) \triangleq \rho_i(\tau,0). \tag{19}$$

In order to obtain useful results on the performance of QDS/SSMA communication over frequency-selective fading channels, an additional assumption about the degree of selectivity of the channel is necessary. For our purposes we shall assume, as in [2], that the channel correlation function satisfies the constraint

$$g_i(\tau) \approx 0 \text{ for } |\tau| > T_d . \tag{20}$$

Using (19) and (20), we find the variance of the diffuse portion of the i-th signal at the output of the in-phase branch of the receiver is

$$\mathrm{Var}\, F_i^{(I)} = \gamma_i^2\, \frac{P}{2} \left\{ \int_0^{T_d} g_i(\tau) \left[|R_{2i}(\tau)|^2 + |\hat{R}_{2i}(\tau)|^2 \right] dt \right.$$
$$\left. + \frac{1}{2} \int_{-T_d}^{T_d} g_i(\tau) \left[|R_{2i-1,2i}(\tau)|^2 + |\hat{R}_{2i-1,2i}(\tau)|^2 \right] dt \right\} , \tag{21}$$

where we have assumed that the data bits for the i-th signal take on value ±1 with equal probability. In a similar manner, using (19) and (20), Var $I_i^{(I)}$ is found to be

$$\mathrm{Var}\; I_i^{(I)} = \frac{PT_c^{\,3}}{24T_d} \sum_{\substack{k=1\\k\neq i}}^{K} (2\gamma_k^2 + 1)[r_{2k-1,2i} + r_{2k,2i}] \,. \tag{22}$$

Using (16), (21), and (22) in (11), $\mathrm{SNR}_i^{(I)}$ for a QDS/SSMA system operating over a frequency-selective fading channel is

$$\mathrm{SNR}_i^{(I)} = \left\{ \frac{N_0}{2\varepsilon} + \gamma_i^2 \frac{1}{T_d^2} \left[2 \int_0^{T_d} g_i(\tau)\left[|R_{2i}(\tau)|^2 + |\hat{R}_{2i}(\tau)|^2\right] d\tau \right.\right.$$
$$+ \int_{-T_d}^{T_d} g_i(\tau)\left[|R_{2i-1,2i}(\tau)|^2 + |\hat{R}_{2i-1,2i}(\tau)|^2\right] d\tau$$
$$\left.\left. + \left[\mathrm{Im}\left\{C_{2i-1,2i}(0)\right\}/N\right\}^2 + \frac{1}{6N^3} \sum_{\substack{k=1\\k\neq i}}^{K} (2\gamma_{\hat{k}}^2 + 1)\left[r_{2k-1,2i} + r_{2k,2i}\right]\right]\right\}^{-\frac{1}{2}} . \tag{23}$$

For a QDS/SSMA system employing quadriphase-coded carriers, (23) may be further simplified to

$$\mathrm{SNR}_{2i}^{(I)} = \left\{ \frac{N_0}{2\varepsilon} + \frac{4\gamma_i^2}{T_d^2} \int_0^{T_d} g_i(\tau) \left[|R_{2i}(\tau)|^2 + |\hat{R}_{2i}(\tau)|^2\right] dt \right.$$
$$\left. + \frac{1}{3N^3} \sum_{\substack{k=1\\k\neq i}}^{K} (2\gamma_k^2 + 1) r_{2k,2i} \right\}^{-\frac{1}{2}} . \tag{24}$$

Once again, although we have presented results only for the in-phase branch of the receiver, with appropriate changes in the subscripts, the above results are also valid for the quadrature branch of the receiver.

NUMERICAL RESULTS

Equations (18) and (24) were numerically evaluated and compared with the results obtained for biphase DS/SSMA from [3] for time-selective fading and frequency-selective fading, respectively. We have tabulated the results for a time-selective fading channel with a triangular correlation function in Table I for two values of $\gamma^2 \triangleq \gamma_j^2$ ($1 \le j \le K$) and for several values of correlation duration (λT_c)[3]. In tabulating these results we have assumed $K = 6$ users employing LSE/AO [7] signature sequences of length $N = 127$ and that $\varepsilon/N_0 = 10\mathrm{dB}$. Further details on the numerical results may be found in [3].

ACKNOWLEDGEMENT

The author wishes to thank F. D. Garber for many helpful discussions on the results presented in this paper.

REFERENCES

[1] P. A. Bello, "Characterization of randomly time-variant linear channels," IEEE Trans. Commun. Sys., pp. 360-393, Dec. 1963.

[2] P. A. Bello and B. D. Nelin, "The effect of frequency fading on the binary error probabilities of incoherent and differentially coherent matched filter receivers," IEEE Trans. Commun. Sys., pp. 170-186, June 1963.

[3] D. E. Borth and M. B. Pursley, "Analysis of direct-sequence spread-spectrum multiple-access communication over Rician fading channels," IEEE Trans. Commun., Oct. 1979.

[4] R. S. Kennedy, Fading Dispersive Communication Channels, Wiley, 1969.

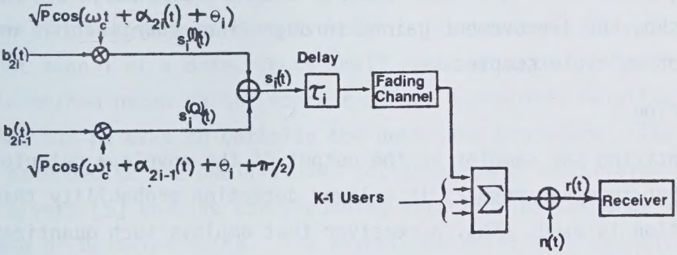
[5] M. B. Pursley and F. D. Garber, "Quadriphase spread-spectrum multiple-access communications," IEEE ICC '78, pp. 7.3.1-7.3.5, June 1978.

[6] M. B. Pursley and F. D. Garber, "Performance of offset quadriphase spread-spectrum multiple-access communications," in prep.

[7] M. B. Pursley and H. F. A. Roefs, "Numerical evaluation of correlation parameters for optimal phases of binary shift-register sequences," IEEE Trans. Commun., Oct. 1979.

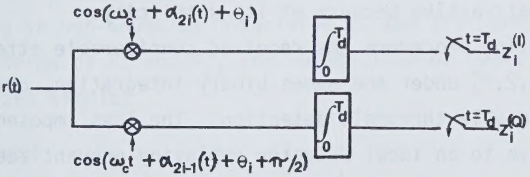
[8] H. L. Van Trees, Detection, Estimation and Modulation Theory, Part III, Wiley 1971.

[9] K. Yao, "Error probability of asynchronous spread-spectrum multiple access communication systems, IEEE Trans. Commun., pp. 803-809, Aug. 1977.



$\sqrt{P}\cos(\omega_c t + \alpha_{2i}(t) + \theta_i)$
$s_i^{(I)}(t)$
$b_{2i}(t)$
Delay
$s_i(t)$
τ_i
Fading Channel
$s_i^{(Q)}(t)$
$b_{2i-1}(t)$
$\sqrt{P}\cos(\omega_c t + \alpha_{2i-1}(t) + \theta_i + \pi/2)$
K-1 Users
$\sum$
$r(t)$
Receiver
$n(t)$

Figure 1A. QDS/SSMA System Model with Fading Channel.



$\cos(\omega_c t + \alpha_{2i}(t) + \theta_i)$
$r(t)$
$\int_0^{T_d}$
$t=T_d$
$z_i^{(I)}$
$\int_0^{T_d}$
$t=T_d$
$z_i^{(Q)}$
$\cos(\omega_c t + \alpha_{2i-1}(t) + \theta_i + \pi/2)$

Figure 1B. QDS/SSMA Receiver for i-th User.

Table I.
SNR(dB)

λ	$\gamma^2=0.1$ DS/SSMA	$\gamma^2=0.1$ QDS/SSMA	$\gamma^2=0.2$ DS/SSMA	$\gamma^2=0.2$ QDS/SSMA
1	11.8	10.83	11.7	10.55
5	11.56	10.47	11.15	9.89
10	11.33	10.11	10.77	9.28
15	11.12	9.79	10.4	8.77
20	10.92	9.50	10.07	8.32
AWGN Channel	11.98	11.17		

OPTIMIZATION OF MULTIPLE-INTEGRATION QUANTIZED-SAMPLE RECEIVERS FOR IMPULSIVE INTERFERENCE ENVIRONMENTS

YOUNGYEARL HAN
RODGER E. ZIEMER
Department of Electrical Engineering
University of Missouri-Rolla
Rolla, Missouri 65401

ABSTRACT

In this paper, the degradation imposed by quantization of the samples at the envelope detector output of a noncoherent receiver is investigated for operation in a type of non-Gaussian interference modeled by Middleton's Class A impulsive noise model. A computerized search procedure is given which results in quantization and decision thresholds for a ternary-integration receiver that are optimized according to the Neyman-Pearson criterion. Because of the difficulty in applying this procedure, the quantization intervals are then chosen so as to maximize the asymptotic efficiency of the receiver. The false alarm and detection probabilities of the receiver are computed for two-, three-, and four-level quantization and integration of 2^m quantized samples with m = 2 through 6. The results show the improvement gained through finer quantization and integration of multiple samples.

Introduction

Quantizing the samples at the output of the envelope detector of a noncoherent receiver results in a lower detection probability than if no quantization is used. Yet, a receiver that employs such quantization and subsequent integration of the quantized samples for comparison with a threshold is attractive because of its simplicity.

This testing procedure has received considerable attention in the literature [1,2,3] under the names binary integration, coincidence detection, and double-threshold detection. The loss imposed by this procedure relative to an ideal detector employing unquantized samples amounts to 1.1 to 1.4dB due to two level quantization for operation in Gaussian noise [1]. In recent years there has been a trend toward designing detectors employing integration of data samples at the envelope detector

---

This material is based upon work supported by the National Science Foundation under Grant ENG77-13878.

output which have been quantized into more than two levels [4,5]. This results in improved detection performance over binary integration.

Gaussian noise is frequently assumed in the analysis of communication systems and this assumption is quite often justified. For certain situations, however, the noise has an impulsive nature. Recently, a statistical-physical model of impulsive non-Gaussian interference has been derived [6,7]. A subclass of this model, referred to as Class A interference, has as a limiting case Gaussian statistics. This model, to be described in greater detail later, will be used in the subsequent analysis to be presented.

In this paper, a noncoherent detector operating in an additive Class A interference environment and employing quantized-sample integration is first optimized according to the Neyman-Pearson procedure. One possible search algorithm is given to find the quantization and decision thresholds that maximize the probability of detection (P_D) subject to a fixed false alarm probability (P_{FA}). Comparison of the detection performance between two- and three-level integration receivers is given.

Because of the difficulty in extending the search algorithm for the optimum thresholds to cases involving the integration of more than four samples, an alternative procedure is sought. It has been shown [8] that if the input signal of a detector is small compared with the noise, then an analytic method based on the concept of the asymptotic relative efficiency (ARE) can be used to optimize the detection procedure. The notion was later extended to the small-signal optimization of multilevel quantization receivers [5] and the corresponding detection performance analyzed for the case of Gaussian noise. The asymptotic-efficiency optimization criterion is applied in this paper to a multilevel quantization envelope detector operating in non-Gaussian interference, and its performance is characterized in terms of P_D and P_{FA} for integration of two-, three-, and four-level quantized samples.

Class A Interference

Impulsive noise can roughly be defined as noise composed of short duration bursts with large peak amplitudes relative to its root-mean-square level. Many models for impulsive noise phenomena have been proposed and used for the design and analysis of signal detectors. A relatively recent model proposed by Middleton [6,7] has the following advantageous features: (1) It is a statistical-physical model in that its parameters are determined explicitly by the underlying physical parameters of the generation mechanism and detector environment; (2) It can be specialized to model interference which is either narrowband relative to the receiver passband (referred to, by Middleton, as class A interference) or wideband relative to the receiver passband (class B interference); (3) It is a canonical model in that mathematical forms do not change with physical conditions. As with several previous models, the received interfering process of Middleton's model is postulated as being composed of a superposition of time translates of a basic waveform. The arrival times of these basic waveforms are assumed to be governed by a Poisson distribution. As such, the form of the first order characteristic function of the noise process corresponding to this superposition of basic waveforms is well known and is dependent on the basic parameter, A, referred to as the impulsive index. The impulsive index can be expressed as the product of the average rate of occurrence of the basic waveforms and the average duration of a typical interfering sample function. Large A corresponds to large overlap among the basic waveforms composing the interference with a corresponding approach to Gaussian statistics, while small A means small overlap and a corresponding highly impulsive nature for the interference.

Impulsive noise is seldom, if ever, the sole interfering noise process in receiving systems. As such, an additional additive independent Gaussian noise component can be included in Middleton's model. The resulting first-order probability density function (pdf), arrived at through a steepest descent approximation and series expansion of the characteristic function, is given by

$$p_Z(z) = \sum_{m=0}^{\infty} e^{-A} \frac{A^m}{m!} e^{-z^2/2\sigma_m^2} \tag{1}$$

where

$$\sigma_m^2 = \frac{m/A + \Gamma'}{1 + \Gamma'} \tag{2}$$

in which Γ' is the ratio of the variance of the Gaussian noise component to that of the impulsive noise component and Z is the noise amplitude normalized by the standard deviation of the sum of the impulsive and Gaussian noise components.

Fig. 1 shows $p_z(z)$ for $\Gamma' = 0.001$ and the various values of A. For small A, $p_Z(z)$ exhibits long tails and a narrow spike at zero amplitude which is characteristic of time domain behavior consisting of large-amplitude short-duration bursts. As A becomes large $p_Z(z)$ approaches as a limiting case a Gaussian distribution. It is shown [6,7] that the Class A interference model can be made to fit well with experimental data by varying the parameters A and Γ'. From the definitions of A and Γ' it is apparent how one would, in principle, determine them through measurement.

The envelope density of narrowband Class A interference plus Gaussian noise plus a sinewave, $\sqrt{2S}\cos\omega_0 t + n(t)$, is given by [6]

$$p_1(v) = \sum_{m=0}^{\infty} e^{-A}\frac{A^m}{m!}\frac{2}{\sigma_m^2} e^{-2S/\sigma_m^2} v e^{-v^2/\sigma_m^2} I_0\left(\frac{2v\sqrt{2S}}{\sigma_m^2}\right), \; v>0 \tag{3}$$

where $I_0(u)$ is a modified Bessel function of the first kind of order zero. This can be reduced to the envelope density of Class A interference plus Gaussian noise, $p_0(v)$, by recalling that $I_0(0) = 1$.

Receiver Optimization and Performance

We consider the case of quantizing the envelope detector output of a non-coherent receiver into several levels as shown schematically in Fig. 2. The receiver input is assumed to be a random-phased sinewave in an additive noise background. The optimization problem is carried out by applying the criteria discussed in the introduction.

At the matched filter output, the noise power is normalized to unity. The v_i's, $i = 1,2,\ldots,n$, are statistically independent envelope samples, and q_1, q_2 and q_3 are quantization thresholds to be determined. The sampled data at the envelope detector output consist of independent samples quantized into, at most, four levels. In what follows H_0 denotes the hypothesis that noise alone is present while H_1 denotes the hypothesis of signal plus noise. The statistic at the integrator output is

$$Z = \sum_{i=1}^{n} y_i \tag{4}$$

which is compared with a decision threshold W. This threshold must be determined such that the required false alarm probability is obtained.

A. Neyman-Pearson Criterion Used to Optimize Thresholds.

For this case we limit the number of quantization thresholds to two. Following Szajnowski [4] it can be shown that the log-liklihood ratio is expressible in the form

$$d = \ell n\, L = n\, \ell n\, (\frac{P_{10}}{P_{00}}) + n_1 c_1 + n_2 c_2 \qquad (5)$$

where

$$c_j = \ell n(\frac{P_{1j} P_{00}}{P_{oj} P_{10}}),\ j = 1,2 \qquad (6)$$

n_j is the number of total samples n falling into quantization interval a_j, and $P_{\ell j}$ is the conditional probability that under the hypothesis H_ℓ the sample v_i will assume a value in the interval a_j.

Since $n\, \ell n\, \frac{P_{10}}{P_{00}}$ is a constant, we can write, equivalently, that

$$d = n_1 c_1 + n_2 c_2 \qquad (7)$$

Since d has a trinomial density, the probability of detection and probability of false alarm become, respectively,

$$P_D = \sum \frac{n!}{n_0! n_1! n_2!} P_{10}^{n_0} P_{11}^{n_1} P_{12}^{n_2} \qquad (8)$$

$$P_{FA} = \sum \frac{n!}{n_0! n_1! n_2!} P_{00}^{n_0} P_{01}^{n_1} P_{02}^{n_2} \qquad (9)$$

The summations are over all n_j such that

$$n = \sum_{j=0}^{2} n_j \quad \text{and} \quad \sum_{j=1}^{2} n_j c_j > W \qquad (10)$$

where W is the decision threshold to be set by the desired false alarm probability.

The probabilities $P_{\ell j}$ can be evaluated by using (3) with S = 0 for $\ell = 0$ and S > 0 for $\ell = 1$. Because of their length they will not be given here, but the interested reader is referred to [9].

A search algorithm for finding quantization and decision thresholds that maximize the detection probability will now be given. We describe the search algorithm in detail for the sample number, n = 3.

For the sample number n = 3, there are ten combinations of n_1 and n_2 for the summation of the trinomial distribution. These ten combinations of n_1 and n_2 are shown in the bracket of Table I. The first number in the parentheses represents n_1 and the second number in the parentheses represents n_2; n_0 can be determined from the equation $n_0=n-n_1-n_2$. Let $c_0=n_1c_1+n_2c_2$ be defined as the value for the combination of n_1 and n_2 in the bracket of Table I. Let $b_0=n_1c_1+n_2c_2$ be defined as the minimum value in the summation. For simplicity, shorthand notation will be used for the summation of the trinomial distribution. For example, the shorthand representation corresponding to row 9 is

$$\sum(3,0)(2,1) = P_{\ell 1}^3 + 3P_{\ell 1}^2 P_{\ell 2}, \ \ell = 0,1$$

Under the condition $c_1 > c_2$, the combinations to the left side of the bracket in Table I are the sure combinations of n_1 and n_2 larger than c_0 and those to the right side of the bracket in Table I are the possible combinations of n_1 and n_2 which can be larger than c_0. This can be easily checked. The search procedure consists of the following steps:

Step 1. Starting the summation from row 10, compute P_{0j}, q_i, and P_{1j}, j=0,1,2; i=1,2 from Eq. (8), (9) and the expressions for the $P_{\ell j}$'s [9] for the given P_{FA}. The first time, the summation contains the left side combination of n_1 and n_2 of the bracket (if these exist) and the combination of n_1 and n_2 in the bracket. If there are no combinations of n_1 and n_2 on the right side of the bracket of the row concerned, find the values of q_1 and q_2 that maximize P_D under the condition $c_1 > c_2$.

Step 2. If there are combinations of n_1 and n_2 on the right side of the bracket of the row concerned, add one more term to the summation and find q_1 and q_2 which gives $c_1 > c_2$. This time, $b_0=n_1c_1+n_2c_2$ is the value for the combination of n_1 and n_2 added in the summation. Exclude the values c_1 and c_2 larger than b_0 for the right side combinations of n_1 and n_2 of the added combination of n_1 and n_2 found in the bracket. Find q_1 and q_2 that maximize P_D.

Step 3. If there is more than one combination of n_1 and n_2 on the right side of the bracket of the row concerned, add one more combination of n_1 and n_2 and follow Step 2. Continue until there is no combination of n_1 and n_2 left in the row concerned.

Step 4. Move up to the next row. Follow the same procedure as above.

The search algorithm was programmed on a digital computer by incrementing P_{0j}, j=0,1,2 by 0.001 holding the equality $\sum_{j=0}^{2} P_{0j}=1$ for the false alarm probability $P_F=10^{-6}$. The resulting curves of probability of detection versus signal-to-noise ratio in dB are shown in Fig. 3 when $\Gamma'=10^{-3}$,

TABLE I

COMBINATIONS OF n_1 and n_2 LARGER THAN c_0 UNDER THE CONDITION $c_1>c_2$ WITH SAMPLE NUMBER, n=3

Row 1									(0,0)			
Row 2	(3,0)	(2,1)	(2,0)	(1,2)	(1,1)	(1,0)	(0,3)	(0,2)	(0,1)			
Row 3			(3,0)	(2,1)	(2,0)	(1,2)	(1,1)	(0,3)	(0,2)	(1,0)		
Row 4						(3,0)	(2,1)	(1,2)	(0,3)	(2,0)	(1,1)	(1,0)
Row 5				(3,0)	(2,1)	(2,0)	(1,2)	(1,1)	(1,0)	(0,3)	(0,2)	
Row 6					(3,0)	(2,1)	(2,0)	(1,2)	(1,1)	(0,3)		
Row 7							(3,0)	(2,1)	(1,2)	(2,0)		
Row 8							(3,0)	(2,1)	(2,0)	(1,2)	(0,3)	
Row 9								(3,0)	(2,1)			
Row 10									(3,0)			

A=10,5,1 and $P_F=10^{-6}$. A comparison of the detection performance between the three-level and the two-level integration receivers also is given in Fig. 3. Although three-level quantization results in better performance than two level quantization, the improvement is not dramatic for the choice of noise and receiver parameters assumed. One is, at this point, naturally curious about the effects of increasing the number of quantization intervals and number of samples integrated. The search procedure just described does not lend itself well to extending the range of these parameters, however. Therefore, application of the concept of asymptotic efficiency (AE) as an optimization criterion for determining the thresholds was investigated.

B. AE Used to Optimize Thresholds.

The AE of a random variable, subject to certain regularity conditions, is given by [8]

$$e(U_n) = \left[\frac{\frac{\partial}{\partial S} E[y] \Big|_{S=0}}{\sigma_0[y]} \right]^2 \tag{11}$$

where

$$E[y] = \int_{-\infty}^{\infty} y\, p(y|H_1)\, dy \tag{12}$$

$$\sigma_0^2[y] = \mathrm{Var}[y|H_0] \tag{13}$$

and y refers to any of the y_i's in (4). The variable S is the signal-to-noise power ratio and the conditions H_1 and H_0, respectively, refer to the signal plus noise and noise only hypotheses.

Following [5], each sample is quantized into an integer according to the transformation $y_i = L-1$ if v_i falls into the interval $[q_L, q_{L+1}]$. By using (3), one may obtain expressions for the density functions $p(y|H_1)$ and $p(y|H_0)$, respectively, for a Class A interference environment. Because of the complexity of these expressions, they are not given here [9]. With these expressions, one may evaluate (11) for the asymptotic efficiency of a single quantized sample at the envelope detector output. It is given by

$$e(U_n) = \frac{T^2}{K-L^2} \tag{14}$$

where

$$K = \sum_{m=0}^{\infty} e^{-A} \frac{A^m}{m!} e^{-q_1^2/\sigma_m^2} + 3\sum_{m=0}^{\infty} e^{-A} \frac{A^m}{m!} e^{-q_2^2/\sigma_m^2} + 5\sum_{m=0}^{\infty} e^{-A} \frac{A^m}{m!} e^{-q_3^2/\sigma_m^2} \tag{15}$$

$$L = \sum_{m=0}^{\infty} e^{-A} \frac{A^m}{m!} e^{-q_1^2/\sigma_m^2} + \sum_{m=0}^{\infty} e^{-A} \frac{A^m}{m!} e^{-q_2^2/\sigma_m^2} + \sum_{m=0}^{\infty} e^{-A} \frac{A^m}{m!} e^{-q_3^2/\sigma_m^2} \tag{16}$$

$$T = \sum_{m=0}^{\infty} e^{-A} \frac{A^m}{m!} \frac{2q_1^2}{\sigma_m^4} e^{-q_1^2/\sigma_m^2} + \sum_{m=0}^{\infty} e^{-A} \frac{A^m}{m!} \frac{2q_2^2}{\sigma_m^4} e^{-q_2^2/\sigma_m^2} + \sum_{m=0}^{\infty} e^{-A} \frac{A^m}{m!} \frac{2q_3^2}{\sigma_m^4} e^{-q_3^2/\sigma_m^2} \tag{17}$$

For two-level quantization, K, L, and T contain only the first summations on the right side of Eq. (15), (16), and (17), respectively, while for three-level quantization, K, L, and T contain the first and second summations of these equations. To find the optimum q_i's, $i=1,2,3$, (i.e., those maximizing the AE for Class A interference), (14) was programmed on a digital computer and the q_i's varied to give a maximum.

To calculate the probability of detection for a preassigned false alarm probability corresponding to the optimum q_i's, we need the pdf of $Z = \sum y_i$. With the assumption of statistical independence of all y_i's, we have

$$p(z|H_\ell) = p(y_1|H_\ell) * p(y_2|H_\ell) * . \; . \; . p(y_n|H_\ell), \; \ell = 0, 1 \tag{18}$$

where the asterisk denotes the convolution operation. For a specific set of quantization thresholds, it is generally impossible to achieve an arbitrary value of P_{FA} because of the discrete nature of random variable Z. In this case, a randomized decision rule can be used to make a fine adjustment to the value of P_{FA}.

The probability of detection for a receiver with quantization thresholds optimized using the AE criterion is shown as a function of signal power S in Figures 4 and 5 for A=10 (nearly Gaussian noise) and A=5 (moderately impulsive noise), respectively. The number of quantization levels and the number of integrations n are employed as parameters while $\Gamma' = 0.001$ and $P_{FA} = 10^{-3}$. Results for A=1 were also obtained but are not shown here.

Discussion

The results shown in Figures 4 & 5 are in agreement with one's intuition in that an increase in the number of quantization intervals or samples integrated results in a decrease in the required S to give a desired P_D. It is also interesting to compare the differences between the values of S required for two- and four-level quantization for a fixed values of P_D and n. Letting these differences be denoted by $\Delta S_{2,4}$ the following approximate values are obtained for $P_D = 0.7$ by reading from the curves of Figures 4 and 5:

	A = 10	A = 5	A = 1
$\Delta S_{2,4}$ for n = 16:	0.7	2.1	1.0
$\Delta S_{2,4}$ for n = 64:	0.7	1.0	0.7

References

[1] M. Schwartz, "A Coincidence Procedure for Signal Detection", IRE Trans. Inf. Theory, Vol. IT-2, pp. 135-139, Dec. 1956.

[2] R. Worley, "Optimum Thresholds for Binary Integration," IEEE Trans. Inf. Theory, Vol. IT-14, pp. 349-353, March 1968.

[3] G. M. Dillard, "A Moving-Window Detector for Binary Integration," IEEE Trans. Inf. Theory, Vol. IT-13, pp. 2-6, Jan. 1967.

[4] W. J. Szajnowski, "A Receiver with Ternary Integration", Proc. IEEE, Vol. 65, pp. 1415-1416, Sept. 1977.

[5] V. G. Hansen, "Weak-Signal Optimization of Multilevel Quantization and Corresponding Detection Performance," NTZ Commun. J., Vol. 22, pp. 120-123, Jan. 1971.

[6] D. Middleton, "Statistical-Physical Models of Man-Made Radio Noise,

Part I: First-Order Probability Models of the Instantaneous Amplitude", Office of Telecommunications, Report 74-36, April 1974.

[7] D. Middleton, "Statistical-Physical Models of Man-Made and Natural Radio Noise, Part II: First-Order Probability Models of the Envelope and Phase", Office of Telecommunications, Report 76-86, April 1976.

[8] J. Capon, "Optimum Coincidence Procedures for Detecting Weak Signals in Noise", 1960 Int. Conf. Rec., pt 4, pp. 154-166.

[9] Youngyearl Han, "Adaptive Multilevel Integration Receivers for Impulsive Interference" Ph.D. Dissertation, EE Dept., Univ. of Missouri-Rolla, May 1979.

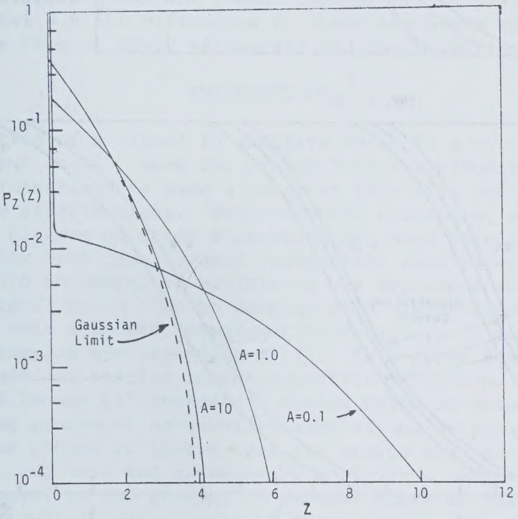


Figure 1. The amplitude distribution of Class A interference for $\Gamma' = 0.001$.

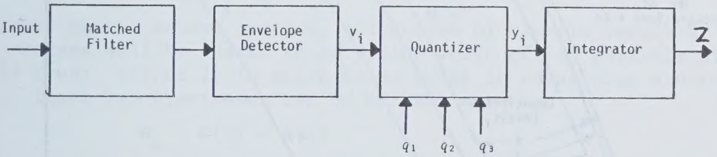


Figure 2. Block diagram of receiver structure.

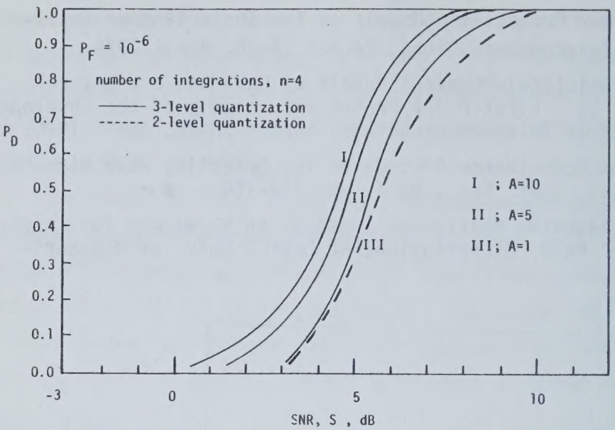


Figure 3

Probability of Detection, P_D, for a 3-level integration receiver operating in Class A interference with $\Gamma' = 0.001$ and $A = 10, 5,$ and 1.

Comparison of P_D for 2- and 3- level quantization for $A = 1$.

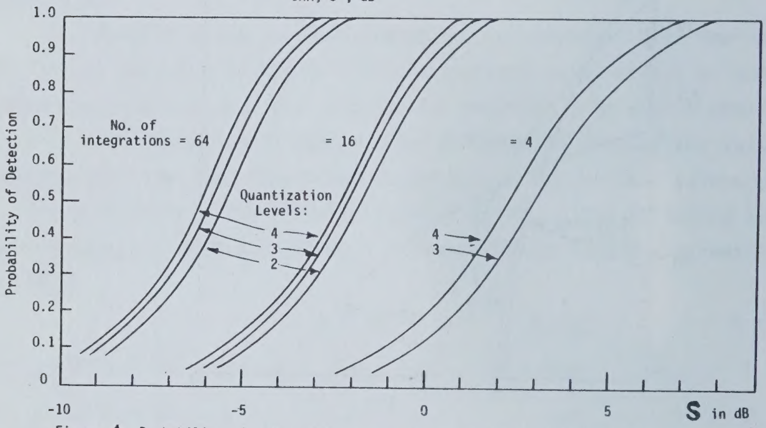


Figure 4. Probability of detection of a receiver optimized by maximizing AE for $P_{FA} = 10^{-3}$, $\Gamma' = 0.001$, and $A = 10$.

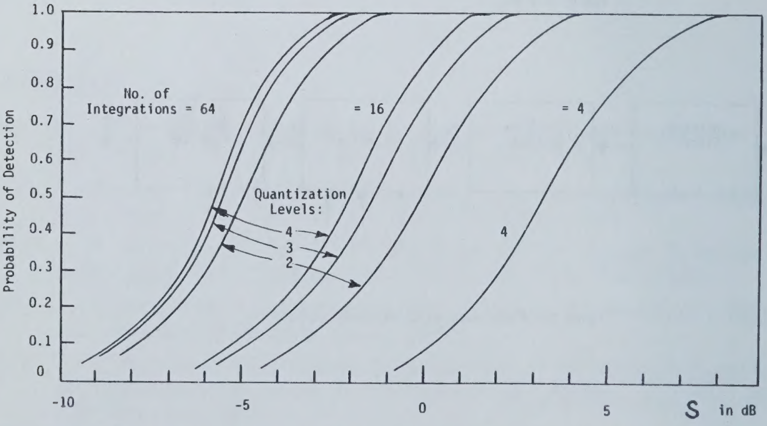


Figure 5. Probability of detection of a receiver optimized by maximizing AE for $P_{FA} = 10^{-3}$, $\Gamma' = 0.001$, and $A = 5$.

NONPARAMETRIC METHODS FOR DETECTION OF KNOWN SIGNALS IN DEPENDENT NOISE

MARC DONVITO AND SALEEM A. KASSAM
Moore School of Electrical Engineering
University of Pennsylvania
Philadelphia, PA 19104

ABSTRACT

Nonparametric methods of detecting signals using dependent samples are studied. A method of taking closely spaced samples and skipping to obtain groups of samples is used. Two new methods of processing these groups, called region tests and linear summing tests, lead to conditional nonparametric tests. The efficacies of these tests are computed, and are used to compare them to other parametric and nonparametric tests.

INTRODUCTION

The detection of a signal in additive noise is a problem that arises in many different areas. When the probability distribution of the noise is unknown, it is necessary to have a detector that will perform well for different noise distributions. Nonparametric detectors, which have a fixed size for large classes of noise distributions, have been studied by many authors, but they have usually used independent samples of the received signal. It would be useful to be able to use dependent samples, since the received signal could then be sampled at a higher rate. However, it has been shown that standard nonparametric detectors do not have fixed size when the samples are dependent [1,2]. This paper investigates methods of handling dependent samples in nonparametric detection.

Kassam and Thomas [3] proposed a method based on obtaining dependent samples, forming groups of consecutive samples, and skipping a sufficient interval between groups to insure that the groups are independent. Their method will be used here and extended by applying a number of different nonparametric tests to the groups. A general class of tests, called region tests, will be studied, and it will be shown that region tests can lead to linear summing tests, which are relatively simple tests based on linear filtering of the groups of samples. These tests are compared to a class of tests proposed by Puri and Sen [4], and are compared to other nonparametric methods using dependent samples.

PROBLEM DEFINITION

Assume that a random process, a function of time, is received. This random process will be tested to determine which of two possible alternatives is true: either it is noise alone or it is noise plus a deterministic signal. These two hypotheses can be written

$$\begin{aligned} H_0 &: X(t) = N(t) \\ H_1 &: X(t) = N(t) + \theta s(t) \end{aligned} \qquad (1)$$

where X(t) is the received signal, N(t) is the noise, s(t) is the deterministic signal, and θ is the signal amplitude. The noise N(t) is assumed to be stationary, and any set of noise samples is assumed to have an absolutely continuous, diagonally symmetric distribution function. This means that for

---

This research is supported by the US Air Force under grant AFOSR 77-3154.

any set of samples $\underline{X}$ of X(t) under H_0, the probability density function (p.d.f.) of $\underline{X}$ satisfies

$$f(\underline{x}) = f(-\underline{x}). \tag{2}$$

The assumption of diagonal symmetry is intuitively reasonable, and allows the construction of tests having fixed size for all noise distributions satisfying the above assumptions.

The detector is implemented by periodically sampling X(t). Suppose that any two samples spaced more than T seconds apart are independent. Such a random process is called T-dependent. By sampling every T seconds or more, independent samples can be obtained. Kassam and Thomas [3] presented a method by which samples can be taken at a higher rate and used in a nonparametric test. Let $r_0 = 1/T$ be the sampling rate required to obtain independent samples. Suppose sampling is done at a higher rate $r = \gamma r_0$, where γ is an integer greater than one. Use p adjacent samples to form one group, skip samples, and form another group of p adjacent samples. Therefore, there is a T-second interval between groups. By forming groups and skipping samples in this way, a set of n groups of p samples is formed. Every group is independent of every other group, but the samples within each group are dependent. Each group can be treated as a multivariate sample point. Henceforth it will be assumed that p-variate groups have been formed in this way.

In some cases, it may not be realistic to assume that the noise is T-dependent. When the noise is not T-dependent, independent groups cannot be formed no matter how many samples are skipped between groups. In such cases the detectors will not have fixed size. It is possible to compute the degradation of the detectors for special cases, using methods similar to those used by Wolff, Gastwirth, and Rubin [1]. However, this problem will not be treated here.

NONPARAMETRIC TESTS

The problem of detecting a signal in noise will be treated as a multivariate hypothesis testing problem. Let $\underline{X} = (X_1, X_2, \ldots X_p)$ represent the p-variate group of samples. Let the vector obtained by sampling s(t) be denoted $\underline{s} = (s_1, s_2, \ldots s_p)$. The vector $\underline{s}$ is assumed to be identical from group to group. This assumption is not necessary, but it simplifies the computations. The case of the known signal, in which $\underline{s}$ is given, will be considered. The two hypotheses can be written

$$\begin{aligned} H_0 &: f(\underline{x}) = f(-\underline{x}) \\ H_1 &: f(\underline{x} - \theta\underline{s}) = f(\theta\underline{s} - \underline{x}) \end{aligned} \tag{3}$$

A test will be called nonparametric if it makes a decision between H_0 and H_1 with fixed size for all allowable probability density functions.

Each set of p-variate measurements $\underline{X}$ can be thought of as being a point in the p-dimensional space of real numbers R^p. We will define the region test to be a test that divides this sample space R^p into a set of regions, assigns a weight to each region, and sums these weights in a specified way to produce a test statistic. Generally conditional tests are required to insure that the region tests are nonparametric. A conditional test is a test in which the threshold and randomization depend on certain values obtained from the sample values [5]. One other requirement must be placed on the region test. Suppose that $P_1, P_2, \ldots P_m, N_1, N_2, \ldots N_m$ are the regions. Then by requiring

$$P\{\underline{x} \varepsilon P_i | H_0\} = P\{\underline{x} \varepsilon N_i | H_0\} \text{ for } i = 1, \ldots m, \tag{4}$$

nonparametric tests can be easily constructed. The property (4) will be called complementarity. It is also possible to have a detector with an odd number of regions, such as a dead zone type of detector. In general the test statistic is

$$T = h(\underline{x}_i) \tag{5}$$

where $h(\underline{x})$ is a weight function defined by

$$h(\underline{x}) = \begin{cases} h \text{ if } \underline{x} \varepsilon P_i \; : \; i = 1, \ldots, m \\ -h \text{ if } \underline{x} \varepsilon N_i \; : \; i = 1, \ldots, m. \end{cases} \tag{6}$$

and $\underline{x}$ is the ith vector of p-variate samples. This statistic T is nonparametric given the number of samples falling into the regions $P_i \cup N_i$, for $i=1,\ldots m$. The number of conditioning parameters is not a function of the dimension of the sample space; it depends only on the number of regions.

The linear summing test is a special type of region test in which the regions are separated by hyperplanes of dimension $p - 1$ (one less than that of the space itself). The simple linear summing test has a weight function

$$h(\underline{x}) = \begin{cases} 1 \text{ if } \underline{a}^t x \geq 0 \\ -1 \text{ if } \underline{a}^t x < 0 \end{cases} \tag{7}$$

where $\underline{a}$ is a constant p-variate vector. The dead zone test has the weight function

$$h(\underline{x}) = \begin{cases} 1 \text{ if } \underline{a}^t x > k \\ 0 \text{ if } -k \leq \underline{a}^t \underline{x} \leq k \\ -1 \text{ if } \underline{a}^t \underline{x} < -k \end{cases} \tag{8}$$

for $k > 0$ given. A multilevel test can be defined by means of the weight function

$$h(x) = \begin{cases} h_i \text{ if } k_i \leq \underline{a}^t x < k_{i-1} ; \; i=1,\ldots,m \\ -h_i \text{ if } -k_{i-1} \leq \underline{a}^t x < k_i ; \; i=1,\ldots,m \end{cases}$$

for $k_0 = \infty$, $k_m = 0$, $k_m < k_{m-1} < \ldots < k_1 < k_0$, and $h_1 > h_2 \ldots > h_m > 0$.

One more type of multivariate test will be considered. It has been constructed and analyzed by Puri and Sen [4], and is a generalization of a bivariate test proposed by Chatterjee [6]. The test statistic is actually a quadratic form. Let the vector $\underline{\tau}$ be defined to have components

$$\tau_i = \sum_{k=1}^{n} \operatorname{sgn}(X_{ik}), \tag{9}$$

where X_{ik} is the ith component of the vector $\underline{X}_k$, which is the kth multivariate group. Let the matrix V have components

$$V_{ij} = \sum_{n=1}^{n} \operatorname{sgn}(X_{ik}) \operatorname{sgn}(X_{jk}). \tag{10}$$

the test statistic is

$$T = \underline{\tau}^t V^{-1} \underline{\tau} \tag{11}$$

The matrix V, which is symmetric with diagonal components equal to n, is the empirical covariance matrix. It can be shown that T is nonparametric given a certain subset of components of V. In particular, the $2_p - 2$ components $V_{11}, V_{12}, \ldots, V_{1p}, V_{23}, V_{24}, \ldots, V_{2p}$ are sufficient to set the threshhold and randomization of T, and no smaller set suffices.

The different nonparametric tests that have been described can be compared on the basis of asymptotic relative efficiency (ARE). According to the Pitman-Noether theorem [5], the ARE is given by the ratio of efficacies for a large class of tests. Thus efficacies will be studied here.

Consider the dead zone region test, which has three regions: P, in which the weight function is +1, DZ, in which it is zero, and N, in which it is -1. The efficacy E_{DZ} of the test statistic can be shown to be

$$E_{DZ} = \frac{2[\int_P \ell(\underline{x}) f(\underline{x}) d\underline{x}]^2}{\int_P f(\underline{x}) d\underline{x}} \tag{12}$$

where $\ell(\underline{x}) = -[\sum_{i=1}^{p} s_i \frac{df}{dx_i}] / f(\underline{x})$. Using a proof similar to the one used by Shin and Kassam [7] it can be shown that this efficacy is maximized by choosing the regions P,DZ, and N to be

$$\begin{aligned} P &= \{\underline{x}: \ell(\underline{x}) > k\} \\ DZ &= \{\underline{x}: -k \leq \ell(\underline{x}) \leq k\} \\ N &= \{\underline{x}: \ell(x) < -k\} \end{aligned} \tag{13}$$

where k is a parameter depending on $f(\underline{x})$. To choose their regions optimally, it is necessary to know $\ell(\underline{x})$, which implies that one must know $f(\underline{x})$. Since, in nonparametric detection, the noise distribution is usually unknown, the region test will be optimized for some nominal noise distribution. The power of the test will be close to optimum when the true noise distribution is close to the nominal distribution, and the size will be fixed under any conditions. Thus the designer is insured against deviations from the nominal noise distribution.

When the nominal distribution is Gaussian, the optimum region test is a linear summing test. The function $\ell(\underline{x})$ can be shown to be

$$\ell(\underline{x}) = \underline{x}^t Q \underline{s} \tag{14}$$

where Q isthe inverse covariance matrix of the multivariate Gaussian density. This test is actually a one-dimensional dead-zone test on the statistic $\ell(\underline{x}) = \underline{s}^t Q \underline{s}$, which is reasonable simple to implement. (Refer to Kassam and Thomas [7] for details concerning the dead-zone test.) Note that $\ell(\underline{x})$ is the log likelihood function used in parametric detection theory [8]. When the dead-zone parameter k of (8) is optimized, the efficacy becomes

$$E_{DZ} = .8101\ \underline{s}^t Q \underline{s}. \tag{15}$$

It is the efficacy of the optimum detector for Gaussian noise multiplied by .8101. For the simple two-level test, the efficacy is

$$E_S = .6366\ \underline{s}^t Q \underline{s}. \tag{16}$$

Computations of efficacy have also been done for region tests with more than three regions. For example, for the four level test, the function $\ell(\underline{x}) = \underline{x}^t Q \underline{s}$ also defines the optimum regions for Gaussian noise, and the efficacy is $.8827 \underline{s}^t Q \underline{s}$ when the region parameter is optimized.

The efficacy of the Puri and Sen tests is a function of the signal vector $\underline{s}$ and the noise distribution. Let $F_{ij}(x_i, x_j)$ be the bivariate noise cdf for the samples i and j, and let f(x) be the univariate noise pdf as defined previously. Define the vector $\underline{\xi}$ to have components

$$\xi_i = 2ns_i f(0), \tag{17}$$

and the matrix Λ to have elements

$$\begin{aligned} \Lambda_{ij} &= n[4F_{ij}(0,0) - 1], \text{ for } i \neq j, \\ \Lambda_{ij} &= n, \text{ for } i = j. \end{aligned} \tag{18}$$

The efficacy of the Puri and Sen test is [4]

$$E_{PS} = \underline{\xi}^t \Lambda^{-1} \underline{\xi}. \tag{19}$$

COMPLEXITY

A conditional test requires the knowledge of certain parameters in addition to the test statistic. For example, for region tests it is necessary to know how many samples fall into each set of complementary regions. For the multivariate sign test of Puri and Sen, it is necessary to know the matrix V. For a conventional unconditional test, there are no extra parameters required. The number of conditioning parameters required can be taken as a measure of the difficulty of implementation of a test, and is defined to be the complexity of the test. Certainly other factors contribute to the difficulty of implementation, and it could be argued that the number of conditioning parameters is not as important as, for example, the number of bits required for a digital implementation of the test. However, by considering complexity as defined here, a number of interesting facts can be stated.

The complexity of the region tests does not depend on the dimension of the multivariate set of samples, nor does it depend on the number of multivariate sets of data used. It only depends on the number of complementary regions used. For the dead-zone region test, the detector has complexity one, since it is necessary to know how many measurements fall into the dead zone. For multilevel region tests, the complexity is the parameter m of equations (4) and (6).

The multivariate tests of Puri and Sen are significantly more complex. As stated previously, 2p-2 components of the matrix V are needed to set the threshold and randomization of these tests. Since one of these components, $V_{11} = n$, is generally known, the complexity might be given by 2p-3. In either case the complexity depends linearly on the dimension of the multivariate sample. This limits the use of these tests to cases in which the group size is small, particularly if the test is implemented by storing the threshold and randomization for all possible conditioning parameters.

NUMERICAL RESULTS

Suppose that the received signal has been sampled, and the samples have been divided into groups with appropriate intervals between the groups. It is worthwhile to study the behavior of different detectors applied to this set of multivariate groups. Numerical comparisons based on the ARE of different tests are shown in Figures 1 and 2. Gaussian noise is used as an example, and a number of nonparametric detectors are compared to the optimal detector for the multivariate groups of samples. All comparisons to be made depend on the knowledge of the signal vector and noise correlation. This information is often known, even when the noise probability density function is not known. Both figures assume a triangular noise covariance function of the form

$$\rho(\tau) = \begin{cases} 1 - |\tau|/T, & \text{when } |\tau| < T \\ 0, & \text{otherwise} \end{cases} \tag{20}$$

and both were calculated using groups of size p = 3. The time between samples within groups is Δ, and the ARE is plotted as a function of Δ/T. This allows comparisons among the detectors to be made for different sampling rates. The known signal is a constant $\underline{s} = (1,1,1)$ in Figure 1, and $\underline{s} = (1,-1,1)$ in Figure 2. The nonparametric detectors shown are the simple linear summing test (based on the sign test), the dead-zone linear summing test, the four level linear summing test, and the Puri and Sen test. It can be seen that the ARE of the linear summing tests with respect to the optimal test for Gaussian noise does not depend on the sampling rate, but for the Puri and Sen it test does. The dead-zone and four level linear summing tests both outperform the Puri and Sen test. In the limit as Δ/T gets large, the correlation between samples decreases. For the triangular covariance function, the ARE of the Puri and Sen test equals $2/\pi$ for $\Delta/T \geq 1$, since the samples are independent. In comparing these tests it must be considered that the Puri and Sen tests are more complicated to implement, but do not require any knowledge of signal and noise parameters.

It is also important to know the effects produced by the sampling and grouping procedure. Tests using different sampling rates and group sizes can be compared on the basis of asymptotic relative processing time (ARPT) [3]. This is defined to be

$$\mathrm{ARPT}_{2,1} = (\mathrm{ARE}_{2,1})\ (T_1/T_2), \tag{21}$$

where $\mathrm{ARPT}_{2,1}$ is the ARPT of detector 2 with respect to detector 1, $\mathrm{ARE}_{2,1}$ is the ARE of 2 with respect to 1, and T_1 and T_2 are the time intervals required by detectors 1 and 2 to obtain a given number of samples. Thus the ARPT can compare two detectors using different sets of samples. It is used here to compare the linear summing detectors to the optimum continuous time (non-sampling) detector. Again Gaussian noise will be used as an example, and it will be assumed that the linear summing tests know the signal shape and noise covariance function. A cosine signal of different frequencies, and noise covariance functions which are the square and cube of the triangular covariance function (20), are used. Figures 3-6 show the ARPT as a function of group size, for the four level test. The ARPT for other linear summing tests is a constant multiple of the points in Figures 3-6; the shapes of the functions in these figures remain the same for other linear summing tests. The ARPT corresponding to group size of one is ARPT of the nonparametric test besed on independent sampling. Summing tests perform significantly better than these independent sampling tests, without requiring large groups. Group sizes of less than twenty will give good performance, if the group size is chosen judiciously. The ARPT values show a cyclical behavior with respect to group size because the skipping interval has been adjusted to begin each group at the maximum of the cosine signal. This insures that the signal vector for each group is the same. (For the independent sampling case, this has not been done. The signal has simply been sampled every T seconds.)

CONCLUSION

Various methods of nonparametric detection of dependent samples have been studied. It was shown that for nominally Gaussian noise and known signal, linear summing tests are the optimal region tests, and have a structure similar to parametric tests. The performance of the multivariate nonparametric test proposed by Puri and Sen has been studied; this test does not require the specification of the signal or a nominal noise distribution. The linear summing tests were compared to the Puri and Sen tests using the ARE, and were shown to be superior when the signal shape and noise covariance function were known to the detector. It was also shown that the implementation of the linear summing tests is relatively simple. The linear summing tests were compared to the optimum continuous time detector using the ARPT. As an example, the detection of a cosine wave in Gaussian noise of different covariance functions was studied. It was shown that good performance can be obtained using groups of less than twenty samples.

REFERENCES

1. S.S. Wolff, J.L. Gastwirth, and H. Rubin, "The Effect of Autoregressive Dependence on a Nonparametric Test," IEEE Trans. Inform. Theory (corresp.), Vol. IT-13, pp. 311-313, April 1967.

2. L.D. Davisson, E.A. Feustal, and J.W. Modestino, "The Effects of Dependence on Nonparametric Detection," IEEE Trans. Inform. Theory, Vol. IT-16, pp.32-41, Jan. 1970.

3. S.A. Kassam and J.B. Thomas, "A Class of Nonparametric Detectors for Dependent Data," IEEE Trans. Inform. Theory, Vol. IT-21, pp. 431-437, July 1975.

4. M.L. Puri and P.K. Sen, Nonparametric Methods in Multivariate Analysis. New York: Wiley, 1971.

5. D.A.S. Fraser, Nonparametric Methods in Statistics. New York: Wiley, 1957.

6. S.K. Chatterjee, "A Bivariate Sign Test for Location," Ann. Math. Statistics, Vol. 37, pp.1771-1782, 1966.

7. J.G. Shin and S.A. Kassam, "Three-Level Coincidence-Correlator Detectors," J. Acoust. Soc. Am., Vol. 63, pp. 1389-1395, May 1978.

8. S.A. Kassam, and J.B. Thomas, "Dead-Zone Limiter: An Application of Conditional Tests in Nonparametric Detection," J. Acoust. Soc. Am., Vol. 60, pp. 857-862, Oct. 1976.

9. H.L. Van Trees, Detection, Estimation, and Modulation Theory, Part I. New York: Wiley, 1968.

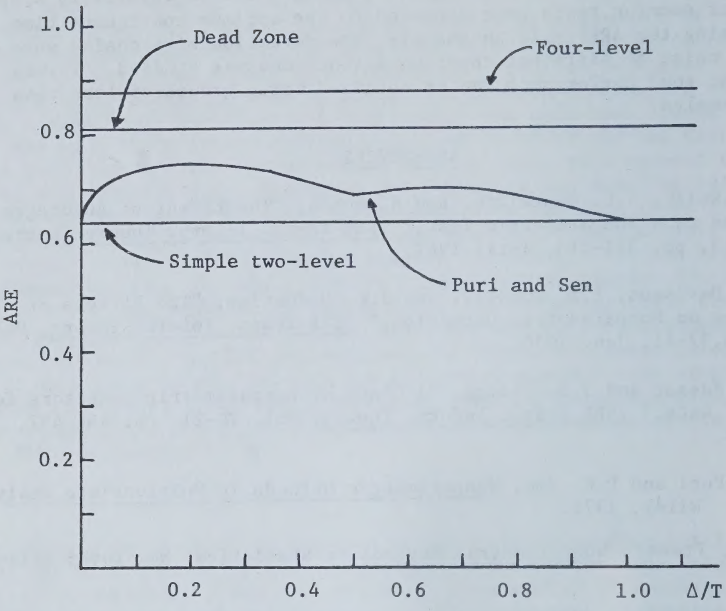


Figure 1. ARE of nonparametric detectors.

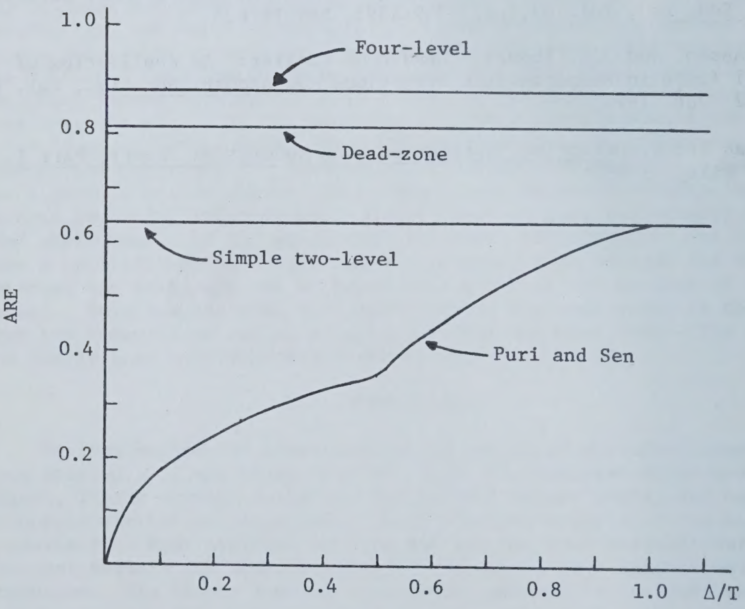


Figure 2. ARE of nonparametric detectors.

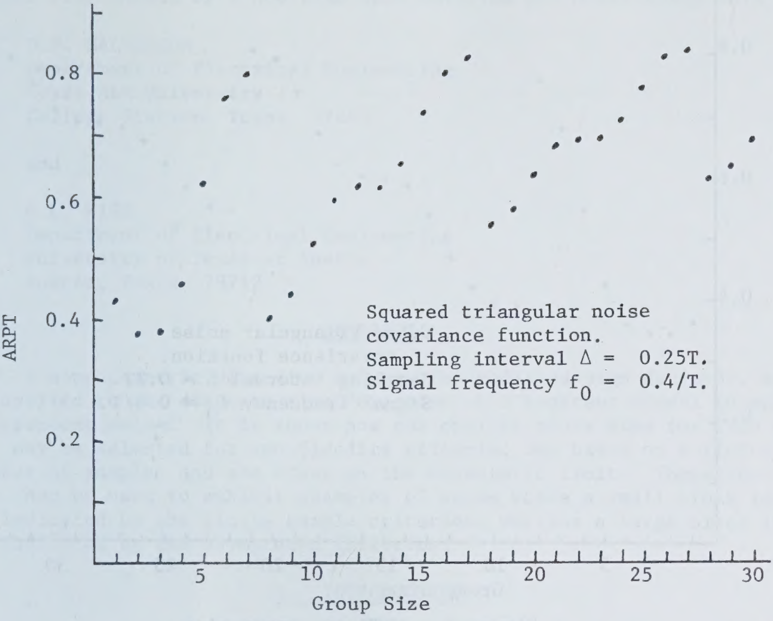


Figure 3. ARPT vs. group size.

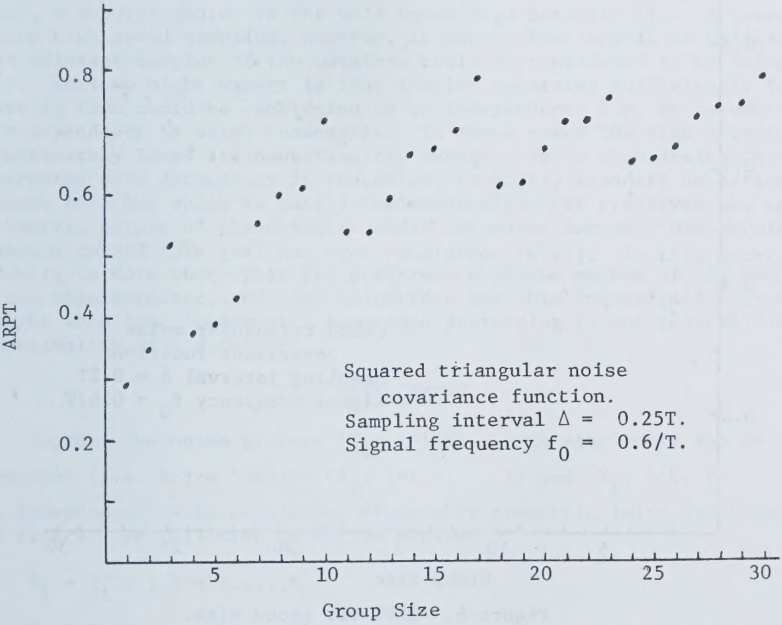


Figure 4. ARPT vs. group size.

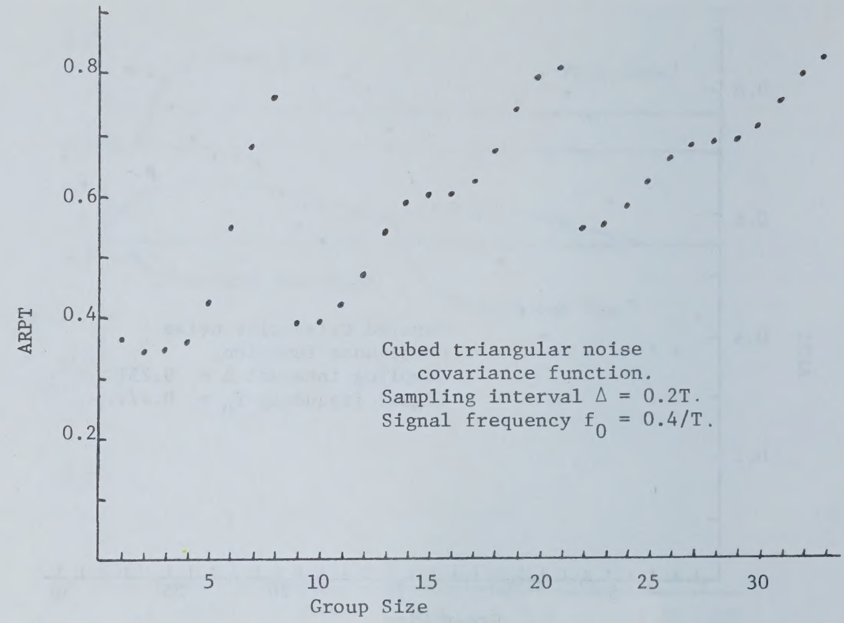


Figure 5. ARPT vs. group size.

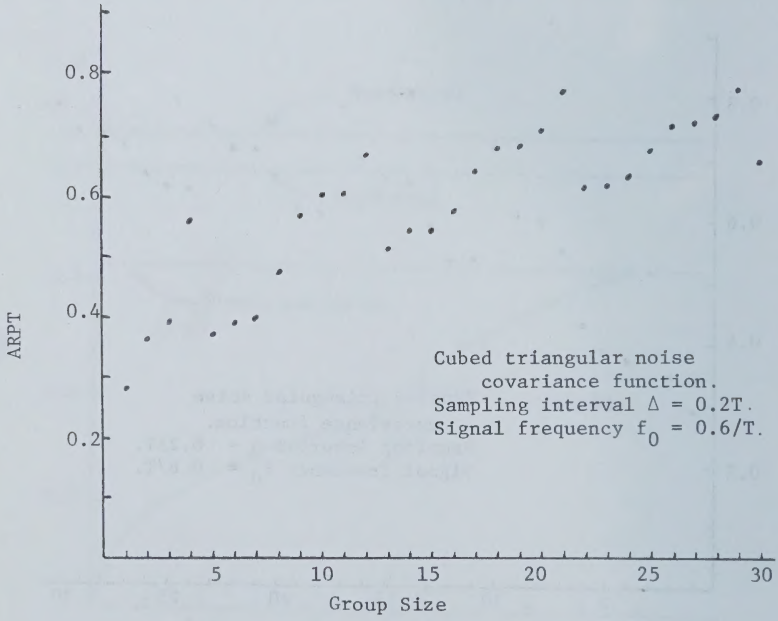


Figure 6. ARPT vs. group size.

ON THE PERFORMANCE OF A MODIFIED SIGN DETECTOR FOR M-DEPENDENT DATA

D.R. HALVERSON
Department of Electrical Engineering
Texas A&M University
College Station, Texas 77843

and

G.L. WISE
Department of Electrical Engineering
University of Texas at Austin
Austin, Texas 78712

ABSTRACT

A nonparametric detection scheme, the modified sign detector, may be applied to the discrete time detection of a constant signal in additive m-dependent noise. It is shown how the optimal block size for this detector may be selected for two fidelity criteria, one based on a finite number of samples and the other on the asymptotic limit. These results may then be used to exhibit examples of cases where a small block length is indicated by the finite sample criterion, whereas a large block length is indicated by the asymptotic criterion.

I. INTRODUCTION

The employment of a nonparametric detector is often desirable in situations where little information about the statistics of the noise is available. If the noise sequence is independent and identically distributed, a popular choice is the well known sign detector [1]. Because of modern high speed sampling, however, in many situations it is unlikely that adjacent samples of the waveform could be considered to be independent. What we might expect is that samples separated sufficiently far apart in time could be considered to be independent, i.e. an assumption of m-dependence is often reasonable. In these cases the sign detector unfortunately loses its nonparametric nature. It is thus desirable, when confronted with dependency in the noise, to modify standard nonparametric schemes in a way which is easily implemented and yet preserves the nonparametric nature of the detector under dependent inputs. One promising approach toward this goal has been considered in [2]. In this paper we investigate more thoroughly the performance of the method of [2] as applied to the sign detector, and also illustrate how this investigation provides insight into some fundamental questions pertaining to the general notion of optimality of a detector.

II. DEVELOPMENT

Suppose the noise process $\{X_i;\ i=1,2,\ldots\}$ is stationary and m-dependent (i.e. $k-j>m$ implies $\{X_i;\ i=1,2,\ldots,j\}$ and $\{X_i;\ i=k,\ k+1,\ldots\}$ are independent) with continuous diagonally symmetric joint densities of all orders. We will wish to decide between

$$H_0:\quad Y_i = X_i \ ;\ i = 1,\ldots,N_0$$

$$H_1:\quad Y_i = X_i+s \ ;\ i = 1,\ldots,N_0$$

where we observe realizations $\{y_i;\ i=1,\ldots,N_0\}$ of the process $\{Y_i;\ i=1,\ldots,N_0\}$ and s is a known positive constant. As applied to the sign detector, the method of [2] requires grouping the samples into blocks of length n with m samples skipped between blocks, resulting in $N_n = G[(N_0+m)/(n+m)]$ independent blocks, where $G(\cdot)$ is the greatest integer function. In general, we will allow fewer than the maximal number N_n blocks and will denote the number of blocks by N. If a summing operation is performed on each block, we obtain a sequence of independent random variables $Z_{1,n},\ldots,Z_{N,n}$ upon which the standard sign detector may be applied. Note that in this case the "modified signal", if present, has strength ns. We will call such a detector a modified sign detector.

A question which naturally arises for this detection scheme is what choice of block length n gives the best performance. The measure of the fidelity of a nonparametric detector has been traditionally vested in employment of the Asymptotic Relative Efficiency (ARE) criterion, which is generally held to be especially appropriate when the signal is weak or the number of observations N_0 is large. When employing the ARE criterion, we often make use of an expression for the ARE given by the Pitman-Noether theorem [3]. In order to apply this theorem, we will need to consider several regularity conditions expressed as follows in terms of the test statistics $T_N(Y_1,\ldots,Y_N)$ which, for the application considered here, may be taken to be zero mean unit variance random variables under H_0 for all N. In the following, $E_s\{\cdot\}$ denotes expectation computed under H_1 with signal strength s and K is a positive constant.

A. $\frac{\partial}{\partial s} E_s\{T_N(Y_1,\ldots,Y_N)\}\big|_{s=0} > 0$

B. $\lim\limits_{N\to\infty} \frac{1}{N}\left[\frac{\partial}{\partial s} E_s\{T_N(Y_1,\ldots,Y_N)\}\big|_{s=0}\right]^2 > 0$

C. $\lim\limits_{N\to\infty} \dfrac{\frac{\partial}{\partial s} E_s\{T_N(Y_1,\ldots,Y_N)\}\big|_{s=K/\sqrt{N}}}{\frac{\partial}{\partial s} E_s\{T_N(Y_1,\ldots,Y_N)\}\big|_{s=0}} = 1$

D. $\lim\limits_{N\to\infty} E_{K/\sqrt{N}}\{(T_N(Y_1,\ldots,Y_N)-E_{K/\sqrt{N}}\{T_N(Y_1,\ldots,Y_N)\})^2\} = 1.$

For a large class of noise processes, the problem of choosing the optimum block length n is addressed by the following:

Theorem 1: Suppose, for the test statistics (where here $N = N_n$)

$$T_{N,n} = \frac{\sum_{i=1}^{N} \operatorname{sgn} Z_{i,n} - N/2}{\sqrt{N/4}}$$

, that conditions A, B, C, D above are satisfied for all fixed choices of block lengths n and some positive constant K. Then the modified sign detector which maximizes the ARE relative to any other such detector has block length n if and only if n maximizes the quantity $\tilde{n}\cdot f_n(0)$, where $f_n(\cdot)$ is the univariate density of $Z_{1,n} = X_1 + X_2 + \ldots + X_n$ and $\tilde{n} = n/\sqrt{n+m}$.

Proof: It follows from the central limit theorem that $T_{N,n}$ converges in distribution to a zero mean unit variance Gaussian random variable as $N \to \infty$, and thus the hypothesis allows application of the Pitman-Noether theorem [3], with the result that an expression for the ARE is given in terms of the efficacy $\eta(\cdot)$. Choice of the optimal block length n thus amounts to choosing n so as to maximize

$$\eta(n) = \lim_{N_0 \to \infty} \left[\frac{\partial}{\partial s} E_s \left\{ \frac{\sum_{i=1}^{N} \operatorname{sgn} Z_{i,n} - N/2}{\sqrt{N/4}} \right\} \Bigg|_{s=0} \right]^2 \Bigg/ N_0 E_0 \left\{ \left(\frac{\sum_{i=1}^{N} \operatorname{sgn} Z_{i,n} - N/2}{\sqrt{N/4}} \right)^2 \right\} .$$

But $\sum_{i=1}^{N} \operatorname{sgn} Z_{i,n}$ is binomially distributed with parameters N and $p_{ns} = \Pr\{Z_{1,n} > 0 | H_1\}$ and thus

$$\eta(n) = \lim_{N_0 \to \infty} \frac{\left[\frac{\partial}{\partial s} N p_{ns} \Big|_{s=0} \right]^2 \cdot \frac{4}{N}}{N_0 \cdot 1} = \left[\frac{\partial}{\partial s} \int_0^\infty f_n(x-ns)\,dx \Big|_{s=0} \right]^2 \cdot 4 \Big/ (n+m)$$

$$= [\widetilde{nf}_n(0)]^2 \cdot 4 .$$

Maximizing $\eta(n)$ thus is equivalent to maximizing $\widetilde{nf}_n(0)$.

QED

If the noise is Gaussian with the "triangular" autocorrelation function, the following corollary to Theorem 1 provides more specific information:

Corollary: Suppose the noise is zero mean Gaussian with autocorrelation

$$R_i = \begin{cases} \frac{m+1-i}{m+1} \sigma^2 , & |i| \leq m \\ 0 , & |i| > m . \end{cases}$$

Then $\widetilde{nf}_n(0)$ increases strictly monotonically for $n \geq 2m$, and thus the optimal block length, as measured by the ARE, is obtained for $n \geq 2m$ by taking n as large as possible.

Proof: It is straightforward to show that the hypothesis of Theorem 1 is satisfied for this noise process. Note that the assumptions on the admissable noise densities imply that the noise is zero mean. Letting σ_n^2 denote the variance of $Z_{1,n}$, we have

$$\sigma_n^2 = n R_0 + 2\{(n-1)R_1 + (n-2)R_2 + \ldots + (n-m)R_m\}$$

$$= \left[n(m+1) - 2 \frac{1 \cdot m + 2(m-1) + \ldots + m \cdot 1}{m+1} \right] \sigma^2 .$$

We then note that $\widetilde{nf}_n(0)$ is proportional to $\frac{n}{\sigma_n \sqrt{n+m}}$, and hence would need to show $h(n) \triangleq \frac{n^2}{(n+m)[n(m+1) - 2 \frac{1 \cdot m + 2(m-1) + \ldots + m \cdot 1}{m+1}]}$

is strictly increasing on $\{n:n\geq 2m\}$, i.e. $\tilde{h}(n) \triangleq n^2/(n^2+an-b)$ is strictly increasing on $\{n:n\geq 2m\}$, where

$$a = m-2\,\frac{1\cdot m+2(m-1)+\ldots+m\cdot 1}{(m+1)^2} \quad\text{and}\quad b = 2m\,\frac{1\cdot m+2(m-1)+\ldots+m\cdot 1}{(m+1)^2}\ .$$

A function of the form of $\tilde{h}$ may be seen to possess a positive derivative for $n > 2b/a$ (it is routine to show $a > 0$). It thus suffices to show

$$n \geq \frac{4m(1\cdot m+2(m-1)+\ldots+m\cdot 1)}{m(m+1)^2-2(1\cdot m+2(m-1)+\ldots+m\cdot 1)}\ ,$$ and hence since $n \geq 2m$, we must show

$$1 \geq \frac{2(1\cdot m+2(m-1)+\ldots+m\cdot 1)}{m(m+1)^2-2(1\cdot m+2(m-1)+\ldots+m\cdot 1)}\ .$$

This may be seen to follow from an induction argument on m.

QED

As a consequence of these results, we would expect that in many cases best performance, as measured by the ARE, can be achieved by employing as large a block size as possible. In the next section we will see how, from a more pragmatic viewpoint, the opposite conclusion may be drawn. The resolution of this apparent disparity will turn out to lie in some limitations associated with the employment of an ARE fidelity criterion.

III. THE FINITE SAMPLE CASE

We have seen that in many cases employment of the ARE fidelity criterion, as applied to the modified sign detector, leads to a choice of block length n as large as possible. From the engineering viewpoint, because of the necessity of utilizing only a finite amount of data, the preferred measure of fidelity would be relative efficiency and not the ARE. The chief reason for employing the ARE is that such an approach is generally more tractable than one based on relative efficiencies. It is generally assumed that if the number of samples is large, whatever can be said about the fidelity of a detector as judged by the ARE also applies if one elects to employ relative efficiency as the measure of fidelity. This is certainly true if the number of samples is sufficiently large, but there is a difficulty in determining what number of samples is sufficient, especially when the true noise statistics are known imperfectly. Because the ARE is a common choice of fidelity criterion for nonparametric detectors, the question of whether or not the ARE consistently reflects the measure of fidelity inherent in the relative efficiency criterion is of genuine concern.

In many cases, we would like to have the design of the detector not depend on a preassigned value of the detection probability β. One way to accomplish this would be to approach the measure of fidelity of the detector through a method which, for a finite number of samples, chooses the block length n so as to maximize β for small signals. This will also turn out to maximize the relative efficiency for certain values of β and signals s. The following theorem will show how this may be accomplished, and will allow addressing the question raised in the previous paragraph.

Theorem 2: Let α be a fixed false alarm probability, and let r_N and t_N be the respective randomization probability and integral threshold required to achieve α for the test statistic

$\sum_{i=1}^{N} \operatorname{sgn} Z_{i,n}$. Then $\left.\frac{d\beta}{ds}\right|_{s=0} = \frac{Nnf_n(0)}{2^{N-1}}\left[(1-r_N)\binom{N-1}{t_N} + r_N\binom{N-1}{t_N-1}\right]$

if $t_N \leq N-1$, and $\left.\frac{d\beta}{ds}\right|_{s=0} = \frac{Nnf_n(0)r_N}{2^{N-1}}$ otherwise.

Proof: Note that t_N and r_N do not in fact depend on n since $\sum_{i=1}^{N} \operatorname{sgn} Z_{i,n}$ is binomially distributed under H_0 independently of n. We have, if $t_N \leq N-1$,

$$\beta = \left[\sum_{i=t_N+1}^{N} \binom{N}{i} p_{ns}^{i} (1-p_{ns})^{N-i} + r_N \binom{N}{t_N} p_{ns}^{t_N} (1-p_{ns})^{N-t_N}\right]$$

where $p_{ns} = \int_0^\infty f_n(x-ns)dx$. Thus if $t_N \leq N-1$,

$$\left.\frac{d\beta}{ds}\right|_{s=0} = \left.\frac{d\beta}{dp_{ns}}\right|_{p_{ns}=\frac{1}{2}} \cdot \left.\frac{dp_{ns}}{ds}\right|_{s=0}$$

$$= \left[\sum_{i=t_N+1}^{N} \binom{N}{i} (i-(N-i))\left(\frac{1}{2}\right)^{N-1} + r_N \binom{N}{t_N} (t_N-(N-t_N))\left(\frac{1}{2}\right)^{N-1}\right] \cdot nf_n(0)$$

$$= \frac{Nnf_n(0)}{2^{N-1}}\left[\sum_{i=t_N+1}^{N} \binom{N-1}{i-1} - \sum_{i=t_N+1}^{N-1} \binom{N-1}{i} + r_N\left\{\binom{N-1}{t_N-1} - \binom{N-1}{t_N}\right\}\right]$$

$$= \frac{Nnf_n(0)}{2^{N-1}}\left[(1-r_N)\binom{N-1}{t_N} + r_N\binom{N-1}{t_N-1}\right] .$$

The desired result, if $t_N \leq N-1$, thus follows. If $t_N = N$ an approach similar to the above completes the proof.

QED

The following lemma will simplify application of Theorem 2:

Lemma 1: For fixed block length n and false alarm probability α, let β_1 be the detection probability obtained from employing N_1 blocks, and β_2 from employing N_2 blocks, where $N_1 > N_2$.

Then $\left.\frac{d\beta_1}{ds}\right|_{s=0} \geq \left.\frac{d\beta_2}{ds}\right|_{s=0}$.

Proof: Since n is fixed, this is equivalent to showing that the classical sign detector for N_1 and N_2 pieces of data respectively has this property. Notice that

$$\left.\frac{d\beta_1}{ds}\right|_{s=0} - \left.\frac{d\beta_2}{ds}\right|_{s=0} = \left(\left.\frac{d\beta_1}{dp_{ns}}\right|_{p_{ns}=\frac{1}{2}} - \left.\frac{d\beta_2}{dp_{ns}}\right|_{p_{ns}=\frac{1}{2}} \right) n\, f_n(0) \;.$$

Since the sign detector is the locally optimal detector for independent stationary Laplace noise, we see that in this situation,

$$\left.\frac{d\beta_1}{dp_{ns}}\right|_{p_{ns}=\frac{1}{2}} \geq \left.\frac{d\beta_2}{dp_{ns}}\right|_{p_{ns}=\frac{1}{2}} \;,$$

since the detector using N_2 samples may be regarded as a detector using N_1 samples (which ignores the extra samples). However, this inequality does not depend upon the assumption of Laplace noise, and we see that it holds in general.

QED

The importance of Lemma 1 is that it allows checking the relevant quantity given by Theorem 2 for fewer combinations of N and n. For each block length n we need only check the corresponding $N_n = G[(N_0+m)/(n+m)]$, which is the largest number of blocks permitted by the data. Note that because of the presence of $f_n(0)$ in the quantity to be checked, we would not in general expect to be able to specify the optimal block length nonparametrically. It would be possible, for example, to estimate $f_n(0)$ by the methods of [4] and [5], which provide pointwise estimates of a density. The great advantage here is that we only need to know the value of the density at the origin, and not elsewhere as in most parametric schemes. We remark that r_N and t_N do not depend on the underlying (symmetric) noise densities.

Suppose now that the noise is 10-dependent zero mean unit variance Gaussian with "triangular" autocorrelation

$$R_i = \begin{cases} \dfrac{11-i}{11} \;, & |i| \leq 11 \\ 0 \;, & |i| > 11 \end{cases} \;,$$

and suppose we have a large fixed number of samples $N_0 = 300$. We will, using Theorem 2 and Lemma 1 find the proper choice of block length n and corresponding N_n which maximizes $\left.\frac{d\beta}{ds}\right|_{s=0}$. To reduce the possibilities to be checked, we first need the following lemma.

Lemma 2: Suppose the noise is Gaussian with autocorrelation

$$R_i = \begin{cases} \dfrac{m+1-i}{m+1}\sigma^2 \;, & |i| \leq m \\ 0 \;, & |i| > m \end{cases} \;.$$

For fixed number of blocks N and false alarm probability α, let β_1 be the probability obtained from employing a block length n_1, and β_2 from a block length n_2, where $n_1 > n_2 \geq m$.

Then $\left.\frac{d\beta_1}{ds}\right|_{s=0} > \left.\frac{d\beta_2}{ds}\right|_{s=0}$.

Proof: From the proof of Theorem 2, it suffices to show $n_1 f_{n_1}(0) > n_2 f_{n_2}(0)$. This follows from methods similar to the proof of the corollary to Theorem 1.

QED

Using Lemma 1 and Lemma 2, it thus suffices to consider only those pairs of n and N given in the following table, where $\left.\frac{d\beta}{ds}\right|_{s=0}$ is computed from Theorem 2 for (n,N):

Table 1. Values of $\left.\frac{d\beta}{ds}\right|_{s=0}$ for Gaussian noise with "triangular" autocorrelation

(n,N)	$\left.\frac{d\beta}{ds}\right\|_{s=0}$
(300,1)	.210
(145,2)	.293
(93,3)	.354
(67,4)	.405
(52,5)	.382
(41,6)	.374
(34,7)	.394
(28,8)	.380
(24,9)	.375
(21,10)	.393
(18,11)	.374
(15,12)	.370
(13,13)	.376
(12,14)	.372
(10,15)	.375
(9,16)	.376
(8,17)	.379
(7,18)	.391
(6,19)	.385
(5,20)	.394
(4,22)	.402
(3,23)	.409
(2,25)	.419
(1,28)	.431

Note that for this example, with 300 samples, the highest detection probability for weak signals is achieved with the smallest block length ($n_1 = 1$), and the lowest detection probability for the largest block length ($n_2 = 300$). However, if one were using the ARE as a measure of fidelity, a choice of large block length ($n \rightarrow 300$) would be indicated by the corollary to Theorem 1. Table 1 thus provides examples of cases where the relative fidelity between two detectors is not preserved as one passes from a finite sample situation to the asymptotic limit. It is likely that one would want to include Gaussian densities, or small perturbations of them, in the nonparametric class considered, and in this case we would know that within this class there exists a nonempty family of densities which exhibit a reversal in relative fidelity as one passes to the asymptotic limit. It is quite common practice to employ the ARE fidelity criterion without attention to the question of how many samples are required to insure that a reversal does not occur. The existence of such a family calls into question the appropriateness of that practice.

Similar difficulties arise when the problem is approached from the viewpoint of relative efficiency. The previous example guarantees that there exist two detectors D_1 (with $n_1=20$) and D_2 (with $n_2=300$) such that for a fixed false alarm probability α, we have $\beta_1 > \beta_2$ for sufficiently small s. If we let $\beta = \beta_1$, then at most 300 samples are required to achieve (α,β) with D_1, but since $\beta_2 < \beta$, Lemma 1 implies that more than 300 samples are required for D_2. Thus the relative efficiency $RE(D_1,D_2; \alpha,\beta,s) > 1$. But $n_1 < n_2$, so the Pitman-Noether theorem, together with the proof of Theorem 1 and its corollary, guarantee that $ARE(D_1,D_2) < 1$. The relative efficiency curve $RE(D_1,D_2; \alpha,\beta,s)$, regarded as a function of s, thus lies above unity for certain values of s before finally crossing unity to converge to the ARE below. Note that the point at which the curve crosses unity corresponds to a reversal in relative fidelity between the two detectors. From the remarks earlier in this paragraph, it is known that at least 300 samples are required before this crossover occurs.

We thus have nonpathological examples of cases for which one might be led to a choice of a large block length n if the ARE criterion is employed, but a small block length if the situation is viewed from the finite sample perspective. This is further evidence to suggest that the difference between relative efficiency and its asymptotic limit is more than simply that of a mathematical definition, and should be considered to be of genuine engineering concern. These results raise questions concerning the appropriateness of universal application of the ARE fidelity criterion as applied to nonparametric detection in dependent noise. Because one must work with a finite amount of data, it would appear that any indication of detector fidelity obtained through the ARE criterion must be interpreted with caution.

IV. CONCLUSION

We have shown how the optimal block size for the nonparametric modified sign detector may be selected for two fidelity criteria, one based on a finite number of samples and the other on the asymptotic limit. We have found by way of example that it is possible for the two criteria to disagree radically on the optimal block size, and that this occurs even when what might be regarded as a large amount of data is available. Viewed in the large, we have produced examples of pairs of nonparametric detectors (D_1,D_2) for which D_1 is the "best" and D_2 the "worst" as judged from the asymptotic viewpoint, but with the opposite situation when judged from the finite sample viewpoint. It would thus appear that cautious employment of the ARE criterion as applied to nonparametric detection in dependent noise would be indicated, with attention given to the minimal number of samples required to insure consistency between the two measures of fidelity whenever possible.

ACKNOWLEDGEMENT

This research was supported by the Air Force Office of Scientific Research, Air Force Systems Command, USAF, under Grant AFOSR-76-3062 and by the Department of Defense Joint Services Electronics Program under Contract F49620-77-C-0101.

REFERENCES

1. J.B. Thomas, "Nonparametric detection," Proc. IEEE, vol. 58, pp. 623-631, May 1970.
2. S.A. Kassam and J.B. Thomas, "A class of nonparametric detectors for dependent input data," IEEE Trans. Inform. Theory, vol. IT-21, pp. 431-437, July 1975.
3. G.E. Noether, "On a theorem of Pitman," Ann. Math. Stat., vol. 26, pp. 64-68, 1955.
4. E. Parzen, "On estimation of a probability density function and mode," Ann. Math. Stat., vol. 33, pp. 1065-1076, 1962.
5. D. Loftsgaarden and C. Quesenberry, "A nonparametric estimate of a multivariate density function," Ann. Math. Stat., vol. 36, pp. 1049-1051, 1965.

DETECTION OF SIGNALS WITH UNKNOWN PHASE

A. H. EL-SAWY
Department of Electrical Communications and Electronics
Helwan University
Helwan, Egypt

ABSTRACT

This paper treats the problem of detection of signals with unknown phase in additive noise in the absence of a mathematical expression for the noise distribution. On the other hand, some partial information about this distribution is assumed to be available. A method which asymptotically guarantees a nontrivial lower bound on performance is derived. In addition, computer simulation results which show the superiority of this system to other available systems at finite sample size is given.

I. Introduction

This paper investigates the detection of signals with unknown phase in additive noise under the assumption that available information about the noise distribution is not complete. Specifically, we shall consider these situations where there is no available mathematical expression for the noise distribution function. On the other hand, it will be assumed that some information is available about the distribution of the inphase and quadrature phase noise components. The signal component will be considered as samples from a demodulated version of the received signal given by

$$S_r(t) = A \operatorname{Sin}(w_c t+\theta) \qquad 0 \leq t \leq T \tag{1}$$

where θ is the unknown phase which is a random variable with uniform distribution over the interval $[-\Pi, \Pi]$, T is the total observation period and A is the signal amplitude which is equal to zero under H_0 and is greater than zero under H_1.

Usually in practice, the engineer uses some version of the square low detector, not because of its optimality but because of its simplicity. In this detection procedure the detector sums the inphase component samples and the quadrature samples, and then utilizes the square root of the sum of the squares of the above two sums as a test statistic. This method has two main defects. First, since the sum is a linear operation, if we have a large noise sample, either because the actual distribution has a long tail or because of some measurement error, then this observation is more likely to upset the decision. Second, if the distribution is not known exactly then setting the threshold will require an adaptive technique. Usually, these adaptive techniques cause some loss in the power of the test [1].

In the following, we give a procedure for the design of robust detectors that guarantee an upper bound on both probability of false alarm and probability of missing under the above mentioned assumptions. We design one of these detectors for the class of p-point distributions, where only one quantile is known. We compare this detector to the square low detector for the lognormal, contaminated normal and Gaussian distributions.

II. Problem Statement

Consider the set of observations $\{z_i\}$ given by its inphase and quadrature phase components denoted by $\{x_i\}$ and $\{y_i\}$ respectively where

$$\begin{aligned} x_i &= A_1 + w_i \\ & \qquad\qquad i = 1, \ldots, N \qquad\qquad (2) \\ y_i &= A_2 + v_i \end{aligned}$$

where $\{w_i\}$ and $\{v_i\}$ are two sequences of independent identically distributed noise which are also mutually independent, $f(w_i)$ and $f(v_i)$ are the density functions of w_i and v_i respectively, which are unknown members of a class of symmetric density functions F, A_1 and A_2 are the inphase and quadrature phase signal components such that $\sqrt{A_1^2+A_2^2} = A$. We define the two hypotheses H_0 and H_1 as

$$\begin{aligned} H_0 &: \; A = 0 \\ H_1 &: \; A > 0 \end{aligned} \qquad (3)$$

We are interested in finding a detection procedure to distinguish between H_0 and H_1 such that both the probabilities of error of the first and second types have the best achievable nontrivial upper bounds.

In the following analysis we shall consider $f(v)$ and $f(w)$ to have the same mathematical expression and we shall deal only with the case where A_1 and A_2 are constants over each group of observations. The generalization to the case where A_1 and A_2 are changing in a known manner as functions of i is straight forward [2].

Before proceeding to the solution, we shall need a few additional definitions. Define a class of functions C such that $L \varepsilon C$ if

1) L is convex, symmetric about the origin and strictly increasing for positive arguments.
2) $l(t) = dL(t)/dt$ is continuous for all t.
3) for all $f \varepsilon F$, $E_f[l^2(x)] < \infty$
4) for all $f \varepsilon F$, $\partial E_f[l(x-\theta)]/\partial\theta$ exists and is non zero in some neighborhood of the origin.

Also, define θ_{NH} as the value of θ such that

$$\sum_{i=1}^{N} l(x_i - \theta_{NH}) = 0 \qquad (4)$$

and

$$\sum_{i=1}^{N} l(y_i - \theta_{NV}) = 0 \qquad (5)$$

and define θ_N as

$$\theta_N = \sqrt{\theta_{NH}^2 + \theta_{NV}^2} \qquad (6)$$

Define also $B_d(0|f)$ and $B_d(A|f)$ as the probability of false alarm and the probability of detection, respectively, using detection strategy d when the actual noise distribution is $f(\cdot)$, and $B_d(\mu|f) = \lim_{N\to\infty} B_d(A|f)$ with $A = \frac{\mu}{\sqrt{N}}$. The subscript L refers to a detector based on a threshold test using θ_N derived from $L \varepsilon C$ as a test statistic; i.e., a test of the form

$$\sqrt{N}\;\theta_N \underset{H_0}{\overset{H_1}{\gtrless}} \gamma$$

we shall call this detector the M-detector on components.

III. Problem Solution

First we require a preliminary lemma whose proof can be found in [3].

Lemma (1):

Whenever $L \varepsilon C$ and $f \varepsilon F$, $\sqrt{N}\ (\theta_{NH}-A_1)[\sqrt{N}(\theta_{NV}-A_2)]$ is asymptotically distributed as a zero-mean, normal random variable with variance

$$V^2(f,L) = \frac{\int_{-\infty}^{\infty} l^2(x)\ f(x)\ dx}{\{\frac{\partial}{\partial\theta} E_f[l(x-\theta)]|_{\theta=0}\}^2}$$

From the above lemma and under the assumption that $\{w_i\}$ and $\{v_i\}$ are independent of each other we obtain with $T_N = \sqrt{N}\ (\theta_N)$,

$$f(T_N|H_0) \underset{N\to\infty}{\to} \frac{T_N}{V^2(f,L)} e^{\frac{T_N^2}{-2V^2(f,L)}} \tag{9}$$

and

$$f(T_N|H_1) \underset{N\to\infty}{\to} \frac{T_N}{V^2(f,L)} e^{\frac{T_N^2+\mu^2}{-2V^2(f,L)}} I_0(\frac{T_N\mu}{V^2(f,L)}) \tag{10}$$

where $I_0(\cdot)$ is the modefined Bessel function of the first kind.

Then the asymptotic power of a test based on T_N satisfies

$$B_L(\mu|f) = \int_{\gamma}^{\infty} \frac{T_N}{V^2(f,L)} e^{-\frac{T_N^2+\mu^2}{2V^2(f,L)}} I_0\ (\frac{T_N\mu}{V^2(f,L)})\ d\,T_N$$

$$= Q(a,\gamma) \tag{11}$$

where $Q(\cdot,\cdot)$ is Marcam's Q function, $a = \frac{\mu}{V(f,L)}$ and γ is the threshold of the test. Notice that for any finite value of γ, if A is a constant then $\mu \to \infty$ as $N \to \infty$ and $Q(\alpha,\gamma) \to 1$; i.e., the test is always consistent.

In the above discussion it was assumed that v_i and w_i are independent for all i. Nevertheless, the results are still correct if v_i and w_i are given by

$$v_i = u_i \cos\phi_i \qquad i = 1,\dots, N \qquad w_i = u_i \sin\phi_i \tag{12}$$

where $\{u_i\}$ and $\{\phi_i\}$ are sequences of independence identically distributed random variables with ϕ_i uniformly distributed between $[0,2\pi]$.

To study the power relations for different $f \varepsilon F$ we need the following lemma.

Lemma (2):

If there exists an $f_0 \in F$ such that $L_0 = -\log f_0 \in C$ then

$$\sup_{L \in C} B_L(\mu|f_0) = B_{L_0}(\mu|f_0) \tag{13}$$

and if

$$V^2(f,L_0) \leq V^2(f_0,L_0) \quad \text{for all} \quad f \in F \tag{14}$$

then the following relation holds asymptotically

$$B_{L_0}(0|f) \leq B_{L_0}(0|f_0) \quad \text{for all} \quad f \in F \tag{15}$$

To get the relation between $B_{L_0}(\mu|f)$ and $B_{L_0}(\mu|f_0)$ we assume that $B_{L_0}(0|f_0) = \alpha$, then

$$B_{L_0}(\mu|f) = \int_{V(f_0,L_0)\sqrt{-2\ln\alpha}}^{\infty} \frac{z}{V^2(f,L_0)} \exp\left[-\frac{z^2+\mu^2}{2V^2(f,L_0)}\right] I_0\left(\frac{z\mu}{V^2(f,L_0)}\right) dz$$

using $R = \dfrac{z}{V(f,L_0)}$

$$B_{L_0}(\mu|f) = \int_{\gamma' = \frac{V(f_0,L_0)}{V(f,L_0)}\sqrt{-2\ln\alpha}}^{\infty} R\, e^{-\frac{R^2+\mu'^2}{2}} I_0(R\mu')\, dR$$

$$= Q(\mu',\gamma') \tag{16}$$

where $\mu' = \dfrac{\mu}{V(f,L_0)}$ and $\gamma' = \dfrac{\gamma}{V(f,L_0)}$.

Notice that as $V(\cdot,\cdot)$ decreases both μ' and γ' increases by the same ratio, which makes this relation untractable analytically, especially with the absence of an analytic expression for the Q function itself. On the other hand, we can rewrite eq. (16) in the form

$$Q(\mu',\gamma') = Q(r\,\bar{\mu}, r\,\bar{\gamma}) \tag{17}$$

where $r = \dfrac{V(f_0,L_0)}{V(f,L_0)} \geq 1$, $\bar{\mu} = \dfrac{\mu}{V(f_0,L_0)}$, and if $\bar{\mu} > \bar{\gamma}$

then [4]

$$Q(r\,\bar{\mu}, r\,\bar{\gamma}) = 1 - e^{-(\mu'^2+\gamma'^2)/2} \sum_{n=1}^{\infty} \left(\frac{\gamma'}{\mu'}\right)^n I_n(\mu'\,\gamma')$$

$$= 1 - e^{-\frac{r^2\bar{\mu}^2}{2}(1+k^2)} \sum_{n=1}^{\infty} (k)^n \, I_n(r^2\bar{\mu}^2 k) \tag{18}$$

where $k = \frac{\bar{\gamma}}{\bar{\mu}} < 1$. Taking the derivative with respect to r,

$$\frac{\partial}{\partial r} Q(r\,\bar{\mu}, r\,\bar{\gamma}) = G[I_1(r^2\,\bar{\mu}^2\,k) - k^2\, I_0(r^2\,\bar{\mu}^2\,k)] \tag{19}$$

where G is a positive constant. Thus

$$\frac{\partial}{\partial r} Q(r\,\bar{\mu}, r\,\bar{\gamma}) \geq 0 \text{ iff } \frac{I_1(r^2\bar{\mu}^2\,k)}{I_0(r^2\bar{\mu}^2\,k)} \geq k^2 .$$ This implies

that if $\bar{\mu}$ and $\bar{\gamma}$ satisfy the above inequality for all values of $r \geq 1$ then the power of the test will be monotonic in r, i.e.

$$B_{L_0}(\mu|f) \geq B_{L_0}(\mu|f_0) . \tag{20}$$

This completes the proof of the following lemma.

Lemma (3):

If there exists $f_0 \varepsilon F$ such that $L_0 = -\log f_0 \varepsilon C$ and

$$V^2(f_0, L_0) \geq V^2(f, L_0) \qquad f \;\varepsilon\; F \tag{21}$$

and if $\frac{I_1(r^2\,\bar{\mu}^2 k)}{I_0(r^2\,\bar{\mu}^2 k)} \geq k^2$ for all $r \geq 1$ and $\bar{\mu} > \bar{\gamma}$ then

$$B_{L_0}(\mu|f) \geq B_{L_0}(\mu|f_0) \qquad f \;\varepsilon\; F . \tag{22}$$

The above inequality was solved for values of $P_F \leq .15$ and it was found that k should be less than or equal to (0.91166). Values of μ are such that the probability of detection is greater than or equal to about 60%. Notice that this is the region of interest in most communication applications. Notice also that in the case of robust coherent detectors [2] this range was for all $P_D \geq .5$.

The above lemma gives the lower bound on the probability of detection over the class F. To investigate how good this lower bound is, we compare the proposed M-detector to the asymptotically most powerful coherent detector when the true noise density is $f_0(\cdot)$. If we assume that the probability of false alarm is α, for the asymptotically most powerful coherent detector

$$B_{opt}(\mu_1|f_0) = 1 - \Phi\,\{\Phi^{-1}(1-\alpha) - \mu_1\, I^{\frac{1}{2}}(f_0)\} \tag{23}$$

where $\Phi(\cdot)$ is the standard normal distribution function and $I(\cdot)$ is the Fisher information number for location. For the proposed M-detector, since $V^2(f_0, L_0) = \frac{1}{I(f_0)}$ then

$$B_{L_0}(\mu_2|f_0) = \int_{\sqrt{-2\ln\alpha}}^{\infty} z\, e^{-\frac{z^2+\mu_2^2 I(f_0)}{2}} I_0[z\,\mu_2\, I^{\frac{1}{2}}(f_0)]\,dz$$

$$= Q(\mu_2\, I^{\frac{1}{2}}(f_0), \sqrt{-2\ln\alpha}) \qquad (24)$$

Comparing the above two equations with the case when $f_0(\cdot)$ is Gaussian [5] we find that if both probability of false alarm and detection are constant then $\frac{\mu_2}{\mu_1}$ is the same in both cases, i.e. in replacing a coherent scheme with an M-detector, the loss in signal to noise ratio will be the same in the M-detector case as replacing linear detector by a square low detector in the Gaussian case. In fact it might seem more appropriate to show that $B_{L_0}(\mu|f_0)$ is greater than or equal to the power of any randomized test of hypotheses under the above assumptions and assuming that the true noise is $f_0(\cdot)$, but with the absence of an optimal procedure to design such a detector, it is the authors opinion that the proposed criterion is the best possible one.

The above can be concluded in the following theorem.

Theorem 1:

If there exists $f_0(\cdot)\varepsilon F \ni L_0 = -\log f_0 \;\varepsilon\; C$ and

$$V^2(f_0, L_0) \geq V^2(f, L_0) \qquad f \;\varepsilon\; F$$

then the M-detector will satisfy the following relations asymptotically

$$B_{L_0}(0|f) \leq B_{L_0}(0|f_0) \qquad f \;\varepsilon\; F \qquad (25)$$

$$B_L(\mu|f) \leq B_{L_0}(\mu|f_0) \qquad L \;\varepsilon\; C \qquad (26)$$

and if $\bar{\mu} = \frac{\mu}{v(f_0, L_0)} \geq \bar{\gamma}$ the threshold and $\frac{I_1(r^2\bar{\mu}\bar{\gamma})}{I_0(r^2\bar{\mu}\bar{\gamma})} \geq (\frac{\bar{\gamma}}{\bar{\mu}})^2$

for all $r \geq 1$

then

$$B_{L_0}(\mu|f) \geq B_{L_0}(\mu|f_0) \qquad f \;\varepsilon\; F \qquad (27)$$

IV. The Family of P-Point Distributions

This section deals with the above detection problem when the noise has a density f which is a member of the family F of p-point distributions defined as

$$F = \{f: \int_{-a}^{a} f(x)d(x) = p,$$

$$f \text{ symmetric and continuous at } \pm a\} . \qquad (28)$$

To apply our procedure to this family, we first find f_0 and L_0 which are given by the following lemma.

Lemma (4):

Over the family F of density functions given by (28), the density function

$$f_0(x) = \begin{cases} b_1 \cos^2(c_1 x), & |x| \leq a \\ b_2 e^{-c_2|x|} & |x| > a \end{cases} \tag{29}$$

which, for appropriate choices of b_1, b_2, c_1, and c_2, is continuous, and has a continuous first derivative, has the following properties: a) $L_0 = -\log f_0 \varepsilon C$, b) $\inf_{L \varepsilon C} V^2(f_0, L) = V^2(f_0, L_0) = 1/I(f_0) = \sup_{f \varepsilon F} V^2(f, L_0)$. The proof of this lemma can be found in either [6],[7] or in [8].

Lemma 4 shows that the detection strategy will be to find θ_{NH} and θ_{NV} such that

$$\sum_{i=1}^{N} l_1(x_i - \theta_{NH}) = 0 \tag{30}$$

and

$$\sum_{i=1}^{N} l_1(y_i - \theta_{NV}) = 0 \tag{31}$$

with

$$l_1(t) = \begin{cases} \tan(c_1 t) & |t| \leq a \\ \tan(c_1 a) \operatorname{sgn}(t) & |t| > a \end{cases} \tag{32}$$

θ_N must then be compared to the threshold γ,

$$\sqrt{N}\, \theta_N \underset{H_0}{\overset{H_1}{\gtrless}} \gamma \tag{33}$$

V. Numerical Results

To examine the finite sample size performance of these M-detectors, we apply the proposed procedure to 3 different distributions

1) Lognormal distribution

$$f(u) = \frac{1}{(2\pi\sigma^2 u^2)^{1/2}} \exp\left[\frac{-2[\ln^2(u)]}{\sigma^2}\right]$$

where $\sigma^2 = 6$ db.

2) Contaminated normal distribution

$$f(u) = (1-r^2)\,\frac{u}{\sigma^2} \exp\left(\frac{-u^2}{2\sigma^2}\right) + \frac{r^2}{g^2\sigma^2} \exp\left(\frac{-u^2}{2g^2\sigma^2}\right)$$

$$+ \frac{2r(1-r)}{g\sigma^2} \exp\left(\frac{-u^2(g^2+1)}{4g^2\sigma^2}\right) I_0\left(\frac{u^2(g^2-1)}{4g^2\sigma^2}\right)$$

Notice that this is the probability density of the envelope of two orthogonal components, each having a density function of the form

$$f(w) = \frac{(1-r)}{(2\pi\sigma^2)^{1/2}} \exp\left(\frac{-w^2}{2\sigma^2}\right) + \frac{r}{(2\pi g^2\sigma^2)^{1/2}} \exp\left(\frac{-w^2}{2g^2\sigma^2}\right)$$

r and g was taken to be 0.25 and 2.25 respectively.

3) Rayleigh distribution

$$f(u) = u\, e^{\frac{-u^2}{2}}$$

Each of the above distributions was considered to be a member of a p-point class of distributions. We set p=.5, then found a. Having a and p the parameters of the nonlinearity $l_1(\cdot)$ in equation (32) were calculated, then equations (30) and (31) were utilized to find θ_N. Using θ_N, a test of the form given by (33) was conducted.

Figures 1 and 2 show the probability of detection P_D verses the signal to noise ration (S/N) for different values of probability of false alarm.

For log-normal (S/N) here is defined as 20 $\log_{10} \frac{A}{M}$, where M is the median of the distribution of the envelope. Figures 3 and 4 show the same for the contaminated normal and the Rayleigh distributions. Notice that the signal to noise ratio required to achieve a certain probability of detection at any fixed value of probability of false alarm in the case of lognormal is higher than that for contaminated normal, which in its turn is higher than the S/N in the Rayleigh case. This is due to the fact that the lognormal has the longest tail and the Rayleigh has the shortest tail. Examining figures (1) and (2) we notice that the M-detector always has a better performance than the square low detector. Notice also that the improvement in S/N becomes better as P_F decreases and achieves about 4 db at 10^{-6}. For the contaminated normal, figure (3) shows that this improvement is between (.5) and 1 db with the M-detector, better than the square low detector. It should be taken into account that the p-point family is not the best family to represent the contaminated normal and much more improvement can be achieved by the utilization of some other distribution classes e.g. [9]. For the Rayleigh distribution, figure (4) shows that the M-detector is about 1 db worse than the square low detector. This loss should be expected since the square low detector is optimal in this case.

An important remark about these similation is that if we normalized the data with the appropriate value of a, then we can set one threshold to guarantee upper bound on the probability of false alarm even at small values of N as N=10. Also, a lower bound on the probability of detection exists. These bounds usually are the threshold and signal required to achieve certain P_F and P_D when the distribution is lognormal. In fact, it was noticed that the changes in P_F and P_D when the distribution of the noise changes from lognormal to contaminated normal are very small.

VI. Conclusion

In this paper we presented a procedure of the detection of signals with unknown phase in noise, under the assumption that the noise distribution is unknown member of a class of symmetric density functions F. It was shown that this method asymptotically guarantees an upper bound on both probability of false alarm and missing whenever the noise density function is element in the class F under consideration. Simulation results were given to show the finite sample size performance of this method and to compare it to the conventional square low detector.

REFERENCES

[1] R. L. Mitchell and J. F. Walker, "Recursive Methods for Computing Detection Probabilities," IEEE Trans. Aerosp. and Electron. Syst. AES-7, pp. 671-676, July 1971.

[2] A. H. El-Sawy and V. D. VandeLinde, "Robust Detection of Known Signals," IEEE Trans. Inform. Theory, Vol. IT-23, pp. 722-727, Nov. 1977.

[3] P. J. Huber, "Robust Estimation of Location Parameter," Ann. Math. Statist., Vol. 35, pp. 73-101, March 1964.

[4] H. L. VanTrees, "Detection, Estimation and Modulation Theory," John Wiley and Sons, N.Y., 1968.

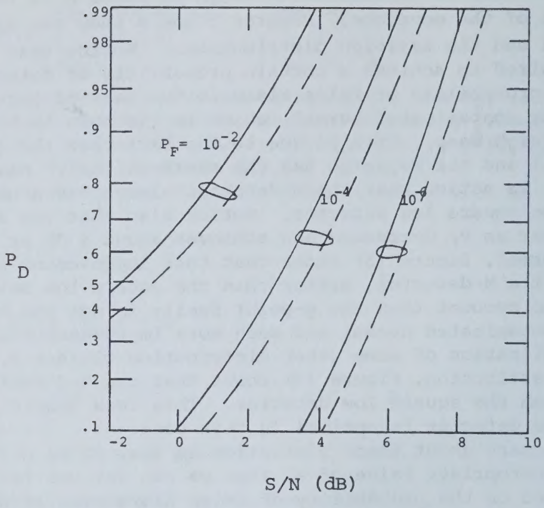
[5] A. D. Whalen, "Detection of Signals in Noise," Acad. Press, NY, 1971.

[6] P. J. Huber, "Fisher Information and Spline Interpolation," Ann. Statist., Vol. 2, No. 5, pp. 1029-1033, Sep. 1974.

[7] E. L. Price and V. D. VandeLinde, "A Solution Technique for the Problem of Minimax Estimation of a Location Parameter," The Johns Hopkins University, Tech. Rep. EE 75-3, 1975.

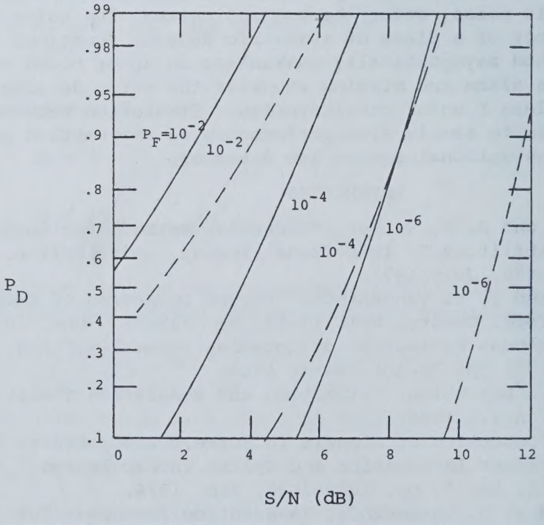
[8] R. D. Martin and C. J. Masreliez, "Robust Estimation via Stochastic Approximation," IEEE Trans. Inform. Theory, Vol.IT-21,pp.263-271, May 1975.

[9] A. H. El-Sawy and V. D. VandeLinde,"Detection of Signals in the Presence of Impulsive Noise," Conference on Inform. Sciences and Systems, Baltimore, MD, March 1978.



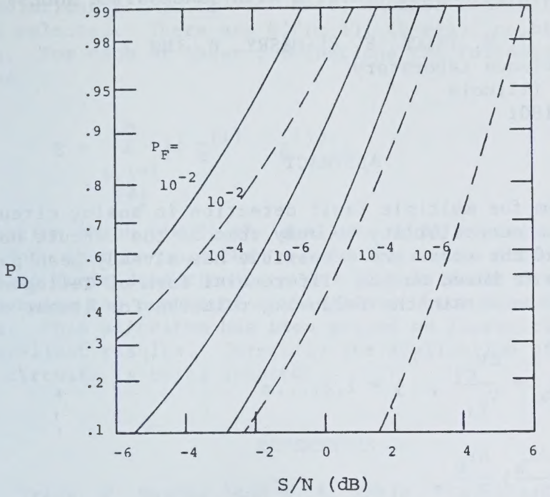
Comparison of square low detector (---) and M-detector on components (—) in log-normal (σ=6dB) interference; N=30

Figure 1



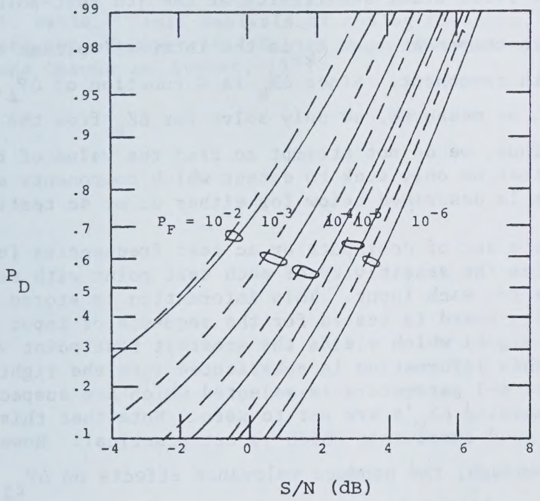
Comparison of square low detector (---) and M-detector on components (—) in log-normal (σ=6dB) interference; N=10

Figure 2



Comparison of square low detector (---) and M-detector on components (—) in contaminated normal interference (r=.25); (g=2.25); N=10.

Figure 3



Comparison of square low detector (---) and M-detector on components (—) in Rayleigh interference; N=10

Figure 4

SINGLE FAULT ANALYSIS OF ANALOG CIRCUITS WITH INACESSIBLE NODES*

A. A. SAKLA**, T. N. TRICK E. EL-MASRY, B. INN
Coordinated Science Laboratory
University of Illinois
Urbana, IL 61801

ABSTRACT

A new algorithm for multiple fault detection in analog circuits is proposed. We assume accessibility to only some of the circuit nodes. The case in which all of the nodes are accessible has already been published [1]. The algorithm is based on the differential form of Tellegen's theorem [2] from which one can obtain the following relation for linear circuits [3].

$$\sum_{k=1}^{b} S_{P_k}^{V_{tj}} \Delta Z_k = \frac{\Delta V_{tj}}{V_{tj}} , \quad j = 1,2,\ldots m \tag{1}$$

where $\Delta Z_k = (1 + \frac{\Delta X_k}{X_k}) \frac{\Delta P_k}{P_k}$,

V_{tj} is the nominal test-point voltage and ΔV_{tj} is the voltage change measured in the faulted circuit, P_k indicates the nominal value of the kth component, $S_{P_k}^{V_{tj}}$ is the first order sensitivity of the jth test-point voltage with respect to the kth component, and X_k is the internal voltage or current associated with the kth component. Since ΔX_k is a function of ΔP_ℓ, $\ell = 1,2,\ldots b$ and cannot be measured, we only solve for ΔZ_k from the test-point measurements. Thus, we do not attempt to find the value of the faulty components, rather we only seek to detect which components are faulty. The algorithm is described below for either dc or ac testing.

First one selects a set of dc inputs or ac test frequencies for a given board and computes the sensitivity of each test point with respect to the circuit components for each input. This information is stored in the test system. The faulty board is tested for the sequence of input test signals, and the test signal which yields the greatest test-point voltage change is selected. This information is substituted into the right side of Eq. (1). Next a set of m-1 parameters is selected which are suspected of being faulty. The remaining ΔZ_k's are set to zero. Note that this assumes zero tolerance on the good components which is not practical. However, if the fault is serious enough, the nonzero tolerance effects on ΔV_{tj} should be negligible. Thus, we have m linear equation in m-1 unknowns. Note also that there are m possible ways to select m-1 of these equations. All m combinations of the m-1 equations are solved to obtain a set of

---

*This work was supported by the Joint Services Electronics Program (U.S. Army, U.S. Navy, and U.S. Air Force) under Contract DAAG-29-78-C-0016, and by the U.S. Naval Air Engineering Center under Contract N68335-78-C-0524.

**A. A. Sakla is now with the Dept. of Electrical Engineering, University of South Alabama in Mobile, Alabama.

possible solutions $\underline{Z}^{(1)}, \ldots \underline{Z}^{(m)}$. Next, another combination of m-1 components is selected. There are b!/(m-1)! (b-m+1)! combinations of m-1 components. For each of these combinations the following error function is computed.

$$E = \sum_{\substack{i,j=1 \\ i \neq j}}^{m} || \underline{Z}^{(i)} - \underline{Z}^{(j)} ||$$

The combination which yields the smallest error contains the faulty components. Note that no more than m-1 faults can be detected, and usually m = 2 (single fault detection), or m=3 (double fault detection), is considered, otherwise the number of combinations which must be examined becomes very large. This algorithm has been tested on linear circuits and has yielded excellent results. Currently the application of the algorithm to nonlinear circuits is being studied.

REFERENCES

1. T. N. Trick, W. Mayeda, and A. A. Sakla, "Calculation of Parameter Values from Node Voltage Measurements," IEEE Trans. on Circuits and Systems, CAS-26, July 1979, pp. 466-474.

2. G. C. Temes and J. W. LaPatra, Introduction to Circuit Synthesis and Design, McGraw-Hill Book Co., New York, 1977, Chapter 9.

3. A. A. Sakla, "Fault Analysis of Analog Circuits," Ph.D. Thesis, Department of Electrical Engineering, University of Illinois at Urbana-Champaign, August, 1979.

GRAPH THEORETIC ASPECTS OF ANALOG FAULT DIAGNOSIS

S. D. BEDROSIAN and J. H. LEE
Department of Systems Engineering
Moore School
University of Pennsylvania
Philadelphia, PA 19104

ABSTRACT

The state of the art in fault analysis of analog networks using graph theory is reviewed briefly and some possible extensions are explored. In particular, the topological interpretation including the network solvability and the key subgraph concept is reviewed and extended.

An algorithm to determine network solvability based on network topology is given. This and other clustering techniques have potential utility in dealing with IC implementations of analog circuits. Some graph theoretic connections with the fuzzy set approach previously presented are also indicated.

I. INTRODUCTION

Two decades of effort on analog fault analysis and diagnosis have failed to produce a unified approach to ATE. The growing complexity of operating electronic systems as well as the extremely large number of components have reemphasized the need for a cost effective systematic approach [6].

We have been applying the fuzzy sets concepts to alleviate some of the difficulties of handling the fault isolation of analog networks in a single fault case [7,8]. Although the above approach with a learning model gave us a general method for determining the best parameters or structures of the fault analysis scheme, it thus far falls short when dealing with the question of multiple fault cases. Such considerations as the convergence property, or the lower and upper limit of the number of the selected features are involved. By investigation of a specific network structure using graph theory and enumeration of the possible number of measurements on linear networks with N-accessible terminals, we can understand better the behavior of faulty analog networks.

Conventional graph theoretical aspects applied for analog fault diagnosis will be briefly discussed [1-4]. Especially the network solvability and key subgraph concepts will be reviewed and extended. The graph-theoretic connection with fuzzy sets for analog fault diagnosis is also indicated.

II. EXISTING GRAPH THEORETIC ASPECTS OF ANALOG FAULT DIAGNOSIS

Applications of graph theory to the analysis and design of electrical networks is not new. But applications of graph theory to network analysis did not give much practical advantage until the advent of the high speed digital computer. Applications of graph theory for fault analysis of combinatorial networks or digital networks have been quite successful [5]. However applications of graph theory to analog fault diagnosis have been less effective. Berkowitz [1] developed the concept of element value solvability for passive linear lumped networks. Bedrosian introduced the key subgraph concept to provide some insight into solutions of active as well as passive lumped networks. Identification of the key subgraph leads directly to a set of equations which in the case of a single element kind network is an homogeneous multilinear algebraic form. Navid and Willson [9] recently presented some sufficient conditions for the element value solvability for linear elements. Here we present a method of determining necessary conditions for network solvability which is tighter than that given by Berkowitz.

III. GRAPH THEORY INVOLVING THREE KINDS OF NODES

We start with a few definitions:

Def. 1) Available nodes (A): External network nodes (terminals) at which voltage and currents can be applied and/or measured. In other words, an available node can be open or short.

Def. 2) Partly available nodes (P): External network nodes at which voltage can be applied and/or measured but currents cannot be applied or measured. In other words, a partly available node can be shorted but cannot be opened.

Def. 3) Nonavailable (inaccessible) nodes (I): Nodes internal to the network at which neither voltages no currents can be measured.

Def. 4) Key subgraph (K): The subgraph of network N which consists of the subset B_k of all branches incident on all the nonavailable nodes.

Def. 5) Core subgraph (C): The subgraph of network N which consists of the subset B_c of all (concealed) branches incident only on the nonavailable nodes.

Using these definitions we state a few theorems without proof.

Theorem 1: For any star graph whose core node is only an I node and at least two of the outside nodes are A, then the network is solvable.

Theorem 2: Consider a given network N partitioned into two subnets N_1 and N_2 along some common nodes. If network N_1 is solvable, you can consider the nodes in N_1 as available nodes except for those common to N_2.

Theorem 3: If a key subgraph is a tree graph and the core subgraph nodes of the tree graph satisfy the following conditions, then the key subgraph is also solvable:(1) each core node which is adjacent to only one other core node is connected with at least 1 A and 1 A or P node respectively. (2) Core subgraph nodes which are adjacent to at least two other core nodes are each connected with at least one A node.

Theorem 3: Consider subnets of network N, N_1 and N_2 both solvable and they share only common nodes. If there is no direct connection among the common nodes, and if it satisfies A(A+2P-1)/2, then the network is solvable.

Corollary. A network N is solvable if the following conditions are all satisfied.

(1) Its key subgraph is solvable.
(2) The network remaining after all the branches adjacent to the core graph C of key subgraph K are deleted is solvable.
(3) The subgraph, which is comprised of key subgraph K enlarged by including all the branches both ends of which are on the key subgraph K, is also solvable.

We note in passing that our Theorem 3 appears to be related to the theorem given recently by Mayeda [10].

IV. ALGORITHM FOR CHECKING NETWORK SOLVABILITY

The following algorithm determines the solvability of the key subgraph K:

1. Find the longest length of core subgraph (tree).
2. Disrupt connectivity (to leave the most favorable branch set).
3. The reduced graph K_r of K generated by following rule is solvable:
   a) Every node of branch connectivity one in the core graph can be changed to the available terminal A if the node has at least two branches connected with available nodes.
   b) Every node of branch connectivity one in the core graph can be changed to partly available nodes.
   c) Every available or partly available node can be divided into two whenever the node is adjacent to the nonavailable nodes.
4. If the reduced graph K_r of K is solvable graph then the key subgraph K is solvable, otherwise go to 5.
5. If we can reduce further go to step 3 and repeat, otherwise stop and K is not solvable.

We include Table I to indicate various facets of the solvability concept. In this paper we concentrated on the key subgraph extensions. In particular we presented a few theorems and an algorithm. We must emphasize that we are mainly concerned with approaches that have potential application to IC implementations of analog circuits.

V. CLUSTERING TECHNIQUES

When one contemplates the diagnosis problem for integrated analog circuits, it becomes clear that some innovations are required to deal with the new set of constraints. Even the idea of "diagnosis to a single faulty component" needs reexamination. The notion of ambiguity groups, i.e., a cluster of several components becomes much more meaningful as a desirable objective.

It turns out that in connection with various other applications a number of clustering procedures have been developed. A convenient summary is presented in Table 2. Note that their corresponding graph theoretic characterizations are also indicated. Many relevant references are also found in the interesting paper by Zahn [11].

Once we open up the possibility of partitioning the IC's into ambiguity groups we must note the likelihood of various degrees of overlap. This again open the door to thinking in terms of fuzzy sets [12]. It is not too surprising to find that additional links exist. The existence of a 1 to 1 correspondence between a class of [0,1] ([0,1,-1]) matrices for ordinary (directed) graphs is well known. The following figure from Kaufmann [13] furnishes another variation; the fuzzy matrix and the corresponding graph representation.

The above calls to mind the efforts at fault detection in the USSR. In particular we note their use of ternary tables to reduce the amount of diagnostic data required to locate faults in discrete devices [14,15]. With this in mind, one is encouraged to seek further leads in recent work by Franksen et al [16]. They discuss system digraph decomposition with a view to establishing broad structural principles underlying controllability and observability in large systems.

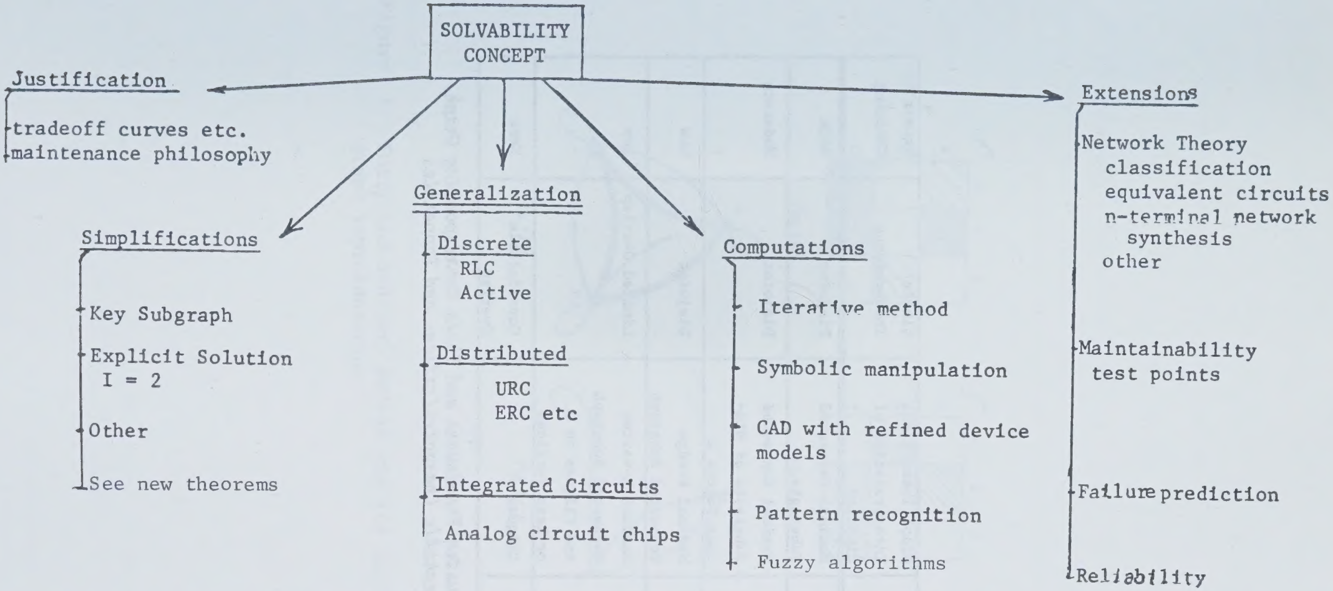

SOLVABILITY CONCEPT
Justification
tradeoff curves etc.
maintenance philosophy
Simplifications
Key Subgraph
Explicit Solution
I = 2
Other
See new theorems
Generalization
Discrete
RLC
Active
Distributed
URC
ERC etc
Integrated Circuits
Analog circuit chips
Computations
Iterative method
Symbolic manipulation
CAD with refined device models
Pattern recognition
Fuzzy algorithms
Extensions
Network Theory
classification
equivalent circuits
n-terminal network synthesis
other
Maintainability
test points
Failure prediction
Reliability

Table 1. An approach focused on the solvability concept

Cluster Procedure	Graph Theoretical Interpretation of Clusters	Cluster Independence	Extent of Chaining
Single Linkage	Maximal Connected Subgraphs	Disjoint	High
K-linkage	Maximal Connected Subgraphs of Minimum Degree k	Disjoint	Moderate
k-edge Connectivity	Maximal k-edge Connected Subgraph	Disjoint	Low
k-vertex Connectivity	Maximal k-vertex Connected Subgraph and Cliques on k or Less Vertices	Limited Overlap	Low
Complete Linkage	Cliques	Considerable Overlap	None

Table 2 - Cluster Procedures and Their Corresponding Graph Theoretic Characterizations and Properties

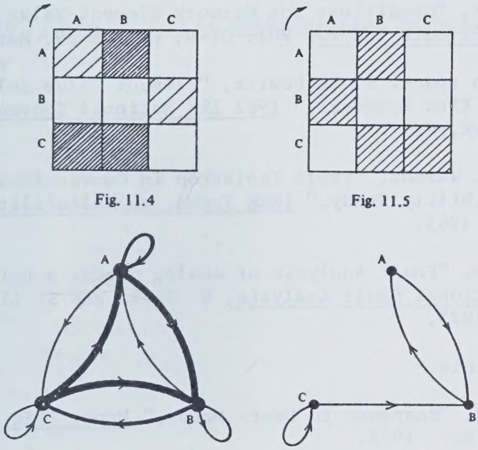

A
B
C
A
B
C
A
B
C
A
B
C
A
B
C
A
B
C

Fig. 11.4 Fig. 11.5

Fig. 11.6 Fig. 11.7

Figure 1. Fuzzy and ordinary matrix and its corresponding graph representation.

Another possible connection that can be explored is that with n × n rectangular arrays or chessboards. Liu [17] discusses the representation of chessboards by rook polynomials. He also indicates their use in dealing with problems involving forbidden positions.

In closing it is clear that much remains to be accomplished.

ACKNOWLEDGMENT

We acknowledge with thanks the research support provided by ONR on Contract N00014-75-C-0768.

REFERENCES

[1] R. S. Berkowitz, "Conditions for Network Element-Value Solvability," IRE Trans. on Circuit Theory, V01. CT-9, pp. 24-29, March 1962.

[2] S. D. Bedrosian and R. S. Berkowitz, "Element Value Solution of Single Element Kind Networks," 1962 IRE National Convention Record Part 5, New York, NY.

[3] S. Seshu and R. Waxman, "Fault Isolation in Conventional Linear Systems-A Feasibility Study," IEEE Trans. on Reliability, Vol. R-15, pp. 11-16, May 1965.

[4] S. D. Bedrosian, "Fault Analysis of Analog Circuits Using Graph Theory," in Rational Fault Analysis, R. Saeks and S. Liberty (eds.), Marcel Dekker 1977.

[5] S. L. Hakimi, ibid.

[6] S. D. Bedrosian, "Response to Users Needs," Proc. IEEE AUTOTESTCON 78, San Diego, Nov. 1978.

[7] S. D. Bedrosian and J. H. Lee, "Application of Fuzzy Set Concepts to Fault Diagnosis," Proc. 22nd Midwest Symposium on Circuits and Systems, Univ. of Pennsylvania, Philadelphia, PA, June 1979.

[8] J. H. Lee and S. D. Bedrosian, "Fault Isolation Algorithm for Analog Electronic Systems Using the Fuzzy Concept," IEEE Trans. on Circuits and Systems, Special Issue on Automatic Analog Fault Diagnosis, CAS-22, pp. 518-522, July 1979.

[9] N. Navid and A. N. Willson, Jr., "A Theory and an Algorithm for Analog Circuit Fault Diagnosis," ibid., pp. 440-456.

[10] W. Mayeda and G. Peponides, "Determination of Component Values in Passive Networks Under Limited Measurements," Proc. Twelfth Asilomar Conf. on Circuits, Systems and Computers, Nov. 1978.

[11] C. T. Zahn, "Graph-Theoretic Methods for Detecting and Describing Gestalt Clusters," IEEE Trans. on Computers, Vol. C-20, pp. 68-86, Jan. 1971.

[12] L. A. Zadeh, "Fuzzy Sets and Their Application in Pattern Classification and Clustering Analysis," in Classification and Clustering, J. Van Ryzin (ed.), Academic Press 1977.

[13] A. Kaufmann, Introduction to the Theory of Fuzzy Subsets Vol. I: Fundamental Theoretical Elements, Academic Press, p. 43, 1975.

[14] Yu V. Malyshenko and A. Kh. Razdobreev, "Method of Reducing the Volume of Diagnostic Information Used for Seeking Faults," Avtomatika i Telemekhanika, No. 4, Transl. pp. 591-599, April 1977.

[15] V. P. Chipulis, "Preprocessing Methods and Formats of Diagnostic Information for Seeking Faults in Discrete Devices," Avtomatika i Telemekhanika, No. 4, Transl. pp. 165-175, April 1977.

[16] O. I. Franksen, P. Falster and F. J. Evans, "Structural Aspects of Controllability and Observability II Digraph Decomposition," J. Franklin Inst., Vol. 308, pp. 105-124, Aug. 1979.

[17] C. L. Liu, Introduction to Combinatorial Mathematics, McGraw-Hill, 1968.

A SINGLE-FAULT DIAGNOSIS THEORY

CHEN-SHANG LIN
R. LIU
University of Notre Dame
Notre Dame, IN 46556

ABSTRACT

A theory for the study of analog circuit single-fault-diagnosis problem is developed in this paper. First, the concepts of fuzzy, precise, and source components along with canonical circuits are introduced. A sufficient condition and an algorithm are then given to determine the diagnosibility of a canonical circuit and the identifiability of the faulty fuzzy-component in the circuit. Finally, an example is included to illustrate the application of the theory.

I. INTRODUCTION

Frequently, a faulty circuit results from very few faulty elements in the circuit. This observation has been used in several papers concerning fault-diagnosis problem of analog circuit. Most of them [1,2,3] employ the "simulation approach". In this approach, the various element values are changed so as to simulate element failures and all the test-point voltages are computed. These results are used either to prepare a "fault dictionary" which would be supplied to diagnosis technicians, or to compare directly with the measured data to determine the faulty elements under some criterion. The computation time and memory size are greatly reduced by the use of very-few-fault observation.

In this paper, the single-fault case is studied. By taking advantage of this information, a sufficient condition based on graph theory is obtained. The process involved a decomposition of a circuit into subcircuits to satisfy certain properties.

A novelty of the paper is the way it is treated for those elements whose nominal characteristics are fuzzily specified, or specified with a large tolerance.

II. PRELIMINARY DEFINITIONS AND ASSUMPTIONS

A circuit component may be a circuit element, a subcircuit or a functional block. Each component is characterized by its characteristic, such as the resistance of a resistor, the gain of an operational amplifier, or the characteristic curves of a transistor.

Definitions.

1. A nominal characteristic is the designed characteristic of the component.

2. An actual characteristic is the characteristic of the component at the time of testing.

3. A precise component, or p-comp, is one whose nominal characteristic is precisely specified. It is fault-free if its actual characteristic is the same as its nominal one.

4. A fuzzy component, or f-comp, is one whose nominal characteristic is not precisely specified, but within a specified range. It is fault-free if its actual characteristic is within this range.

5. A source component, or s-comp, is one whose terminal-voltages and terminal-currents can be measured at the time of testing.

Examples of fuzzy components are transistors, operational amplifiers and electrolytic capacitors.

Remarks.

1. For fault diagnosis problems, the nominal characteristics are usually given and the actual characteristics are unknown.

2. In a realistic case, the actual characteristic of a precise component may deviate from its nominal one under fault-free case, but the deviation is small, say less than five percent.

Assumptions.

1. The actual characteristic of a component can be uniquely determined from its terminal-voltages and terminal-currents.

2. The nominal characteristic of a p-comp is voltage-controlled, i.e., its terminal-currents is uniquely determined by its terminal-voltages.

Definitions.

6. A circuit is diagnosable w.r.t. a set of measurements if all faulty components can be located from the measurements.

7. A component of a circuit is identifiable w.r.t. a set of measurements if its actual characteristic can be determined from the measurements.

III. THE MAIN THEOREM

In this section, a restricted class of circuits will be considered. It is a class of circuits crucial to one-component-fault problems. The extention to the general circuit will be considered in the later section.

Definition 6. A circuit is canonical if it satisfies the following:

(C1) All components are two-terminal.

(C2) It contains only p-comp and f-comp.

(C3) Its graph is non-separable.

(C4) There is neither f-comp only loops, nor p-comp only cutsets.

(C5) There is at most one component connecting any two nodes.

The canonical circuit may be excited by any number of s-comps, connecting across any two nodes, with any waveforms, as long as Assumption 1 is satisfied. With the above condition, the following theorem can be stated.

Theorem 1. A canonical circuit is diagnosable and its faulty f-comp is identifiable, if

(A1) at most one component is at fault.

(A2) all node voltages are measurable.

Proof: The proof of Theorem 1 is based on the following algorithm.

Step 1. Calculate the terminal currents for all p-comps, from their nominal characteristics and measured node voltages. These currents are called the estimated currents.

Step 2. Calculate the terminal currents for all f-comps, from the estimated currents obtained in Step 1. This is done by the fundamental cutset equations [4,5] associated with the tree containing all and only f-comps. The existence of such a tree is guarenteed by Condition (C4). These obtained currents again are called the estimated currents.

Step 3. Determine the characteristic of f-comps from their estimated currents and measured node voltages, called the estimated characteristic.

Step 4. Label each f-comp with the letter "T" or "F", according to the following rule. If the estimated characteristic is within its specified range, label it "T", otherwise, "F". There may be zero, one, or more than one f-comps labeled F. In the last case all f-comps labeled F have to be in the same fundamental loop.

Step 5. Locate the fault component according to the following rule.

1. If there is no F-labeled f-comp, then there is no faulty component in the circuit.

2. If there is one F-labeled f-comp, then this f-comp is at fault.

3. If there is more than one F-labeled f-comps, then the p-comp which defined the fundamental loop containing these f-comps is at fault.

It remains to be shown that the decision made in Step 5 is a correct one. Since there are at most one faulty components, there are only three possibilities.

1) There is no faulty components. Therefore, the estimated currents of all p-comps are actual currents. As a consequence, all estimated currents and estimated characteristics of f-comps are actual ones. Therefore, there will be no F-labeled f-components.

2) There is one f-comp at fault. In this case, all p-comps are fault-free. By the same reason as above, all estimated currents and estimated characteristics of f-comps are actual ones. Therefore, there will be one F-labeled f-comp.

3) There is one p-comp at fault. In this case the estimated current of this, and only this, p-comp is at error. As a consequence, all the estimated currents are at error for those, and only those, f-comps in the fundamental loop defined by this p-comp. Therefore, their estimated characteristics will also be at error, causing a switch in their labelling from T to F. Since, in view of Condition (C5), there are at least two f-comps in every fundamental loop, there will be two or more f-comps labeled by F. This completes the proof that the circuit is diagnosable.

If one f-comp is at fault, from the above discussion, all estimated characteristics of f-comps are the actual ones. In particular, the actual characteristic of the faulty f-comp can be obtained. Therefore, it is identifiable. The theorem is proved.

The definition of canonical circuit and the conditions of Theorem 1 seem to be very restrictive. Actually this is not the case. The only crucial condition is that there exists no f-comps only loops and that there is at most one component at fault. The others can be alleviated by a proper grouping of circuit elements into components. This is illustrated by the next example.

IV. AN EXAMPLE

An example will be given which apparently does not satisfy Conditions (C1), (C4) and (A2).

Example 1. Consider a voltage regulator as shown in Figure 1. The measurable node voltages are at nodes (1), (2), (3) and (4), and the s-comps are at the input and output terminals. From these measurements, we want to locate the one-element-fault.

Note first that node voltages at nodes (A) and (B) are not measurable. This can be alleviated by grouping elements (Q1,Q2,D2,R5) into a 4-terminal f-comp and (D3,D4) into a two-terminal f-comp. The other f-comps are

D1 and C. The rest of them are p-comps. Next, the above 4-terminal f-comp is further represented by three coupled two-terminal f-comps. The resulting circuit is shown in Figure 2. Note that all node voltages are now measurable.

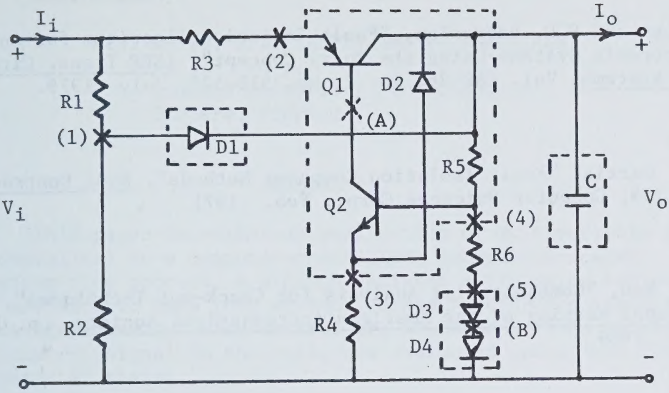
In order to check if the circuit is canonical or not, the s-comps are first removed. It is seen that there is a cutset of P1 and P3. This situation can be alleviated in the following way. Calculate the estimated currents for P1 and P3, and see if the cutset equation of P1, P3 and S1 is satisfied or not. If it is not satisfied, then the fault is clearly at either P1 or P3. No further calculations need to be done. If the cut set equation is satisfied, then both P1 and P3 are fault-free. In addition, the estimated currents are the actual currents. Therefore, P1 and P3 become s-comps. The remaining circuit no longer has any P-comp, only cutsets.

Since P3 is now considered as a s-comp, when removed the component F2 is separable from the circuit. A close check, the actual current of F2 can also be found. Therefore, it can also be considered as a s-comp. The resulting circuit is shown in Figure 3, which is canonical except that components F2, F3 and F4 are coupled.

By the same reasoning a partial answer can be obtained and it is summarized in Table 1. Note that only the partial labels are used for the decision and the coupling relation between (F2, F3, F4) is not used.

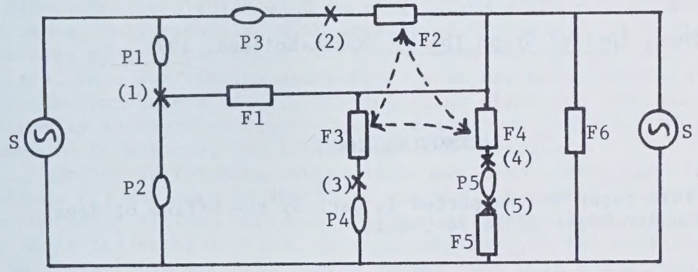
Observed Labels			Elements at Fault
F1	F5	F6	
T	T	T	(F2, F3, or F4), or no fault
T	T	F	F6 or P4
T	F	T	F5
T	F	F	P5
F	T	T	F1
F	T	F	P2
F	F	T	F5
F	F	F	More than one fault

Table 1. Diagnosis Diagram for Example 1.



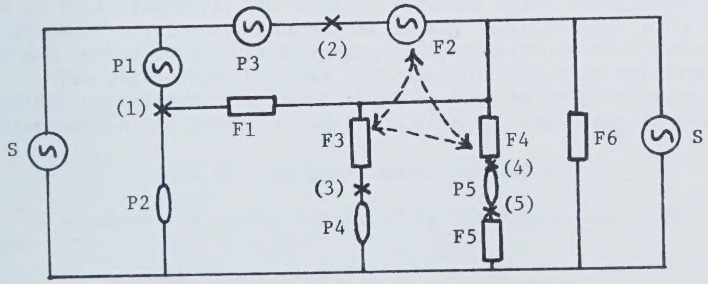
I_i
+
R3
(2)
Q1
D2
I_o
+
R1
(A)
(1)
D1
R5
V_i
Q2
(4)
C
V_o
R6
(3)
(5)
D3
R4
R2
(B)
D4
-
-

Figure 1



P3
(2)
F2
P1
(1)
F1
S
F3
F4
F6
S
(4)
P2
(3)
P5
(5)
P4
F5

Figure 2



P3
(2)
F2
P1
(1)
F1
S
F3
F4
F6
S
(4)
(3)
P5
P2
(5)
P4
F5

Figure 3

REFERENCES

[1] J. Lee and S.D. Bedrosian, "Fault Isolation Algorithm for Analog Electronic Systems Using the Fuzzy Concept", IEEE Trans. Circuits and Systems, Vol. CAS-26, No. 7, pp. 518-522, July 1979.

[2] R.F. Garzia, "Fault Isolation Computer Methods", NASA Contract Rep., CR-1758, Computer Sciences Corp., Feb. 1971.

[3] E.C. Neu, "Combinatorial Analysis for Check-out Techniques", Joint National Meeting of the American Astronautical Society, pp. 17-20, June 1969.

[4] S. Seshu and M.B. Reed, Linear Graphs and Electrical Networks, Addison-Wesley 1961.

[5] W. Chen, Applied Graph Theory, North-Holland, 1971.

ACKNOWLEDGEMENT

This paper was supported in part by the Office of Naval Research under Grant N0014-74-C-0444.

EXPERIMENTAL EVALUATION OF MULTIVARIABLE SELF-TUNING CONTROLLERS *

H.N. KOIVO, J.T. TANTTU AND J. PENTTINEN

Tampere University of Technology
Tampere, Finland

ABSTRACT

This paper describes an application of multivariable self-tuning controllers to a computer-controlled, laboratory-scaled concentration-flow process. A minimum variance self-tuning controller, which is modified to take into account constant reference signals, and a newly developed self-tuning controller, which is derived by penalizing also the control signal in the cost, are evaluated using both simulated and experimental tests.

INTRODUCTION

In practice, the control engineer most often does not build a careful model for a plant in order to design a controller for it, but uses a PID-controller, which with careful tuning works sufficiently well in most cases. Only recently, with developing technology, have significant advances been made in the basic DDC-scheme with the idea of self-tuning controllers. In a self-tuning controller, when new measurements are obtained, the parameters of a fixed controller structure are updated using normally a recursive least-squares algorithm. The model of the plant is assumed to be unknown, but linear and time-invariant.

Scalar self-tuning controllers have been widely used in several applications [1] and have proved to be extremely powerful. Suggestions to improve the basic minimum variance self-tuning controller have been made in [2], where the control signal is also penalized in the cost function. In the multiple-input multiple output (MIMO) case only a few results are available ([3], [4], [5],).

In [3] the minimum variance self-tuning regulator has been extended to a multivariable case. It often produces, however, excessive peaking in the control signal. This can be reduced by having a quadratic control term in the cost function. The corresponding self-tuning controller structure has been derived in [5]. Set-point changes, as well as adding an integrator in the loop to remove the steady-state offset, are easily handled in this framework. The minimum variance self-tuning regulator, on the other hand, requires modifications in the basic scheme. Both of these methods are tested on a laboratory-scaled concentration-flow process.

The paper first reviews the relevant theory of multivariable self-tuning controllers. The pilot plant is then briefly described. Simulation results are presented next and finally, experimental results.

REVIEW OF MIMO SELF-TUNING CONTROLLER

Consider the system described by the linear vector difference equation:

$$A(q^{-1})y(t) = B(q^{-1})u(t-k) + C(q^{-1})\xi(t) \tag{1}$$

where $y \in \mathbb{R}^m$ is the output, $u \in \mathbb{R}^m$ is the control, $\xi \in \mathbb{R}^m$ and $\{\xi(t)\}$ is a sequence of independent equally distributed random vectors with zero

---

*This research has been supported in part by the Finnish Academy.

mean value and covariance $E\{\xi(t)\xi^T(t)\} = R$, k is the time delay, and q^{-1} is the backward shift operator, $q^{-1}y(t) := y(t-1)$. A, B, and C are polynomial matrices in q^{-1} with $A(o) = C(o) = I$, $B(o) = B_o$, B_o nonsingular, and $\det C(z)$ has all its zeros outside unit disk.

The structure of the self-tuning controller is obtained by minimizing the cost function

$$I = E\{||y(t+k) - w(t)||^2 + ||\,Q'(q^{-1})\,u(t)||^2\} \qquad (2)$$

subject to (1). Here $w(t) \in \mathbb{R}^m$ is the reference signal and $Q'(q^{-1})$ chosen to be either λI or $\lambda(1-q^{-1})I$ where λ is a tuning parameter.

The details of a minimization procedure can be found in [5]. Resulting optimal control law is given by

$$\tilde{F}(q^{-1})y(t) + \tilde{G}(q^{-1})u(t) + \tilde{H}(q^{-1})w(t) = 0 \qquad (3)$$

where the matrices $\tilde{F}$, $\tilde{G}$, and $\tilde{H}$ are determined from the matrix equations (dropping q^{-1})

$$\tilde{F}\,E = \tilde{E}\,F$$

$$\tilde{G} = \tilde{C}^{-1}\,EB + Q$$

$$H = -\tilde{C}$$

and

$$C = AE + q^{-k}\,F$$

$$\tilde{C} = \tilde{E}A + q^{-k}\,\tilde{F}$$

$$Q = (B_o^T)^{-1}Q'^T(0)\;Q'$$

For the self-tuning controller it is first assumed that $C(q^{-1}) = I \Rightarrow C(q^{-1}) = I$. Then define vectors

$$\phi^*(t+k|t) := y^*(t+k|t) - w(t) + Q(q^{-1})u(t)$$
$$= \tilde{F}(q^{-1})y(t) + \tilde{G}(q^{-1})u(t) + \tilde{H}(q^{-1})w(t)$$

and

$$\phi(t+k) := y(t+k) - w(t) + Q(q^{-1})u(t)$$

where $y^*(t+k|t)$ signifies the k-step ahead predictor.

Define the data vector $x(t)$ by

$$x(t) := [y^T(t),\ y^T(t-1),\ldots;u^T(t),\ u^T(t-1),\ldots;$$
$$w^T(t),\ w^t(t-1),\ldots\ ] \qquad (4)$$

and the parameter matrix Θ by (dropping ~ in the sequel)

$$\Theta = [\theta_1,\ldots,\theta_m] := [F_0,\ F_1,\ldots,\ G_0,\ G_1,\ldots,$$
$$H_0,\ H_1,\ldots\ ]^T$$

where the column vector θ_i is given by

$$\theta_i = [f^0_{i1},\dots,f^0_{im}, f^1_{i1},\dots, f^1_{im},\dots; g^0_{i1},\dots, g^0_{im}, g^1_{i1},\dots, g^1_{im},\dots; h^0_{i1},\dots, h^0_{im}, h^1_{i1},\dots, h^1_{im},\dots]^T$$

$$i = 1,\dots, m,$$

The control $u(t)$ is computed from (3) to be

$$G_0\, u(t) = -[\sum_{i\geq 0} F_i\, y(t-i) + \sum_{i\geq 1} G_i\, u(t-i) + \sum_{i\geq 0} H_i\, w(t-i)] \tag{5}$$

The control parameters Θ are estimated by a standard recursive least-squares algorithm in the manner indicated in [3]

$$\hat{\theta}_i(t+1) = \hat{\theta}_i(t) + K(t)\, [\phi_i(t) - x(t-k)\hat{\theta}_i(t)] \tag{6}$$

$$i = 1,\dots, m,$$

$$K(t) = P(t)x^T(t-k)[1 + x(t-k)P(t)x^T(t-k)]^{-1} \tag{7}$$

$$P(t+1) = \{P(t) - K(t)[1 + x(t-k)P(t)x^T(t-k)]K^T(t)\}/\beta \tag{8}$$

Here β is the exponential forgetting factor, the values of which should be between $0.9 < \beta \leq 1$.

Computationally it is more advantageous to use the square-root algorithm [6] for updating the parameters. Rounding errors occur, especially in computers (or microprosessors) with short word length, making it possible for the positive definite P-matrix to loose this property.

Algorithm ($Q = \mathrm{diag}\,\{\lambda_1,\lambda_2\}\, I$, where $\lambda_i, i=1,2$, is a constant parameter)

1. Read new output $y(t)$, setpoint $w(t)$.
2. Compute $\phi(t) = y(t) - w(t-k) + Q\, u(t-k)$.
3. Form data vector $x(t-k)$ (4).
4. Update $\hat{\theta}_i$ by recursive least-squares algorithm (possibly in square-root form): (6)-(8).
5. Generate new control from (5):

$$\hat{G}_0\, u(t) = -[\sum_{i\geq 0} \hat{F}_i\, y(t-i) + \sum_{i\geq 1} \hat{G}_i\, u(t-i) + \sum_{i\geq 0} \hat{H}_i\, w(t-i)]$$

6. Set $t = t + 1$ and go to 1.

The (modified) algorithm of [3] is obtained simply by setting $\lambda_1=\lambda_2= 0$. Constant reference values occur frequently in process control, when e.g. the system is changed from one set-point to another.

The controller structure (5) easily handles this special situation. If the steady-state error is to be removed, an integrator can be introduced by changing the cost function slightly. In that case $Q = \lambda(1-q^{-1})I$, where only one tuning parameter λ is chosen for simplicity.

The only point of departure in the algorithm is in step 2. This would now read:

2. Compute $\phi(t) = y(t) - w(t-k) + \lambda(u(t-k) - u(t-k-1))$.

Another way of introducing an integrator in the loop is given in [8]. The multivariable case does not change the situation essentially. Everywhere where $y(t)$ occurs, Wittenmark uses $y(t) - w(t)$, and instead of $u(t)$, he uses $u(t)-u(t-1)$, if the system does not have integrators.

DESCRIPTION OF THE PROCESS

The process is illustrated in Fig. 1. For a more comprehensive description of the process and its instrumentation see [7]. The lower tank is the storage tank, which is divided into two parts by a separating filter so that in the lower part there is concentrated water-plastic powder stock and in the upper part thin stock. The stocks are pumped into the mixing tank above by d.c.-motordriven pumps. The mixture flows back from the mixing tank into the lower part of the storage tank and from there some of it further flows through the separating filter into the upper part. The measuring quantities are the concentration and the rate of flow from the mixing tank, and these are also the quantities to be regulated. The process is controlled by regulating the rotating speeds of the feed pumps. The control voltages of the pumps are in the range 0 - 10 V. The measurement signals of the concentration and the rate of flow are voltages in the range of 0 - 10 V.

Typically, the parameters of the process are slowly time-varying at the beginning. The process also has some mild nonlinear characteristics.

SIMULATION RESULTS

By various identification techniques the linearized model for the process at an operating point has been verified to be

$$y(t) + A_1 y(t-1) = B_0\, u(t-1) + \xi(t) + C\, \xi(t-1), \tag{9}$$

where

$$A_1 = \begin{bmatrix} 0.98 & 0 \\ -0.01 & 0.98 \end{bmatrix}, \quad B_0 = \begin{bmatrix} 0.012 & 0.012 \\ -0.023 & 0.023 \end{bmatrix},$$

$$C = \begin{bmatrix} 0.02 & 0.7 \\ -0.9 & -0.2 \end{bmatrix}.$$

Note especially that B_0 has quite small elements which can cause numerical difficulties.

In the computer simulation of the discrete model of the pilot plant the self-tuning controller with tuning parameter $\lambda = 0$ (the same value for λ_1 and λ_2 was used) worked quite satisfactorily. The outputs followed well step-wise set-point changes, but the variances of the control signals were quite large, as can be seen in Fig. 2. In real processes such variances are not acceptable and it turned out that with $\lambda=0$ the controller failed, when applied to the pilot plant. The effect of the tuning parameter λ is quite apparent when Fig's 2 and 4 are compared. The variances of control signals are much smaller when $\lambda=0.1$ is used. The effect of λ on output signals is not very large but noticeable, however. In Fig. 4 a constant error between the mean value of y_2 and the reference signal w_2 can be observed.

In computer simulations the choice of the initial values of parameters and variance matrix P(0) had no noticeable effect on controller performance except for the values of matrix G_0. It is a good choice to set other parameters zero at t=0, but G_0 should be invertible. Usually G_0 was initially chosen to be an identity matrix. The initial value of the variance matrix P(t) was 100 I in Fig's 2-4.

As a rule of thumb, a value kI, where $10 \leq k \leq 100$ is a good choice when the square-root algorithm is used. The only crucial parameter choices appear to be the values of the tuning parameters λ_i, which determine the trade-off between variances of control and output signals.

EXPERIMENTAL RESULTS

The MIMO self-tuning controller algorithm was tested using the described pilot plant. Fig's 5-7 contain the essential features obtained from the experiments.

In Fig. 5 the plant was driven from the equilibrium position to a set-point $y_r=[7.2,7.2]$. The tuning parameter λ was chosen to be 0.25 and the initial P(0)=10,000I, which was higher than used in the simulation. The response is quite good (Fig. 5 (a)), although a slight steady-state error occurs. This can be improved by using two tuning parameters λ_1 and λ_2, but it requires more experiments. The parameters converge quite rapidly fairly close to the "correct" values of eq. (9) as is seen from Fig's 5 (b)-(d).

The same experiment was also performed with $\lambda=0$, but this resulted in a failure: the controllers were oscillating from one limit to another without being able to control the outputs. In order to reduce the steady-state error to zero an integrator was attached to the algorithm in the way suggested above. Although it resulted in partial success, the output response was quite oscillatory and the settling time extremely long compared with the proposed controller. It is possible that there exists a value for λ giving satisfactory response, but in any case the tuning was much more difficult. In simulations, the difference between the basic controller and the one with an integrator was not significant. In Fig. 6 the idea is to see the behavior of the proposed controller, when the set-points are different. Only the outputs and the control are shown. Essentially the same results were obtained as in Fig. 5.

Finally, Fig.7 illustrates the characteristics of the proposed controller, when several set-point changes occur. The time-scale here is five times longer than in Fig's 5 and 6. The system was first driven from the equilibrium position to a set-point [7.2,7.2], exactly as in Fig. 5. Then at t=250 s, the new set-point for the flow is 6.4 and at t=500s, the set-point of the concentration also becomes 6.4. At t=750s, the new set-point for concentration is 7.2. As can be seen from Fig.7, the controller performs quite well and, in fact, its performance improves the further time elapses. The parameters converge again quite rapidly and this time to almost the same values as given in eq. (9).

CONCLUSIONS

In this paper a MIMO self-tuning controller was tested on a laboratory-scaled concentration flow process, first by simulation and then by performing experimental tests. The model of the pilot-plant has been obtained by standard identification schemes (e.g. by generalized least-squares method).

The simulation studies indicated that the proposed MIMO self-tuning controller of [5] should work quite satisfactorily, but some problems might occur when $\lambda=0$. This then proved to be the case in performing experiments with the pilot plant. When λ was equal to zero, which corresponds to a modified Borisson's controller, the controller failed completely. With appropriate tuning of λ, quite reasonable results were obtained. The most satisfying case was the one of Fig. 7, when several set-point changes occurred and the controller improved its performance, when time elapsed.

The results are still preliminary, but prove that the MIMO self-tuning controller of [5] has several promising features. It does not require careful modeling of the plant. Set-point changes are easily accommodated in the controller. The previously suggested minimum-variance self-tuning controllers of [3] and [4], which essentially correspond to $\lambda=0$, did not work with the pilot-plant. The points that require further study are:What are the reasons for the failure of the controller with an integrator to work as well as in the simulation. Are there efficient ways to choose tuning parameters. Can the controller be used to tune PI-controllers efficiently? Although the name - self-tuning - would indicate that tuning is not required, this is not true. There are a number of parameters that have to be tuned, but once the controller has been tuned on, it does improve its performance and performs well in noisy and even nonlinear situations.

REFERENCES

[1] Åström,K.J., Borisson, U., Ljung, L., and B. Wittenmark (1977). Theory and applications of self-tuning regulators. Automatica 13, 457-476.

[2] Clarke, D.W., and P.J. Gawthrop (1975). Self-tuning controller. Proc. IEE 122, 929-934.

[3] Borisson, U. (1979). Self-tuning regulators for a class of multivariable systems. Automatica 15, 209-215.

[4] Keviczky, L., Hetthéssy,J., Hilger, M., and J. Kolostori (1978). Self-tuning adaptive control of cement raw material blending. Automatica 14, 525-532.

[5] Koivo, H.N.(1979). A multivariable self-tuning controller. Submitted to Automatica. Also available as Report 31, Tampere Univ. of Tech.,Dept. of Elect. Eng., Tampere, Finland.

[6] Peterka, V. (1975). A square root filter for real-time multivariable regression. Kybernetika 11, 53-67.

[7] Rajala, M. (1976). A pilot plant for concentration control. Sähkö-Electricity in Finland 10, 385-389.

[8] Wittenmark, B. (1973). A self-tuning regulator. Report 7311, Dept. of Automatic Control, Lund Inst. of Tech., Lund, Sweden.

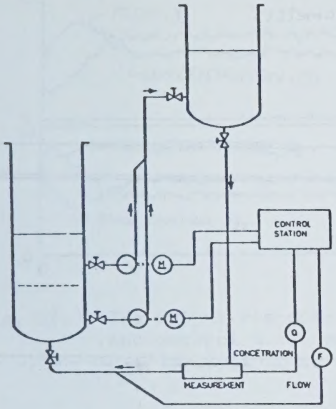

CONTROL STATION
M
M
Q
F
CONCETRATION
MEASUREMENT
FLOW

Fig. 1. The concentration-flow process.

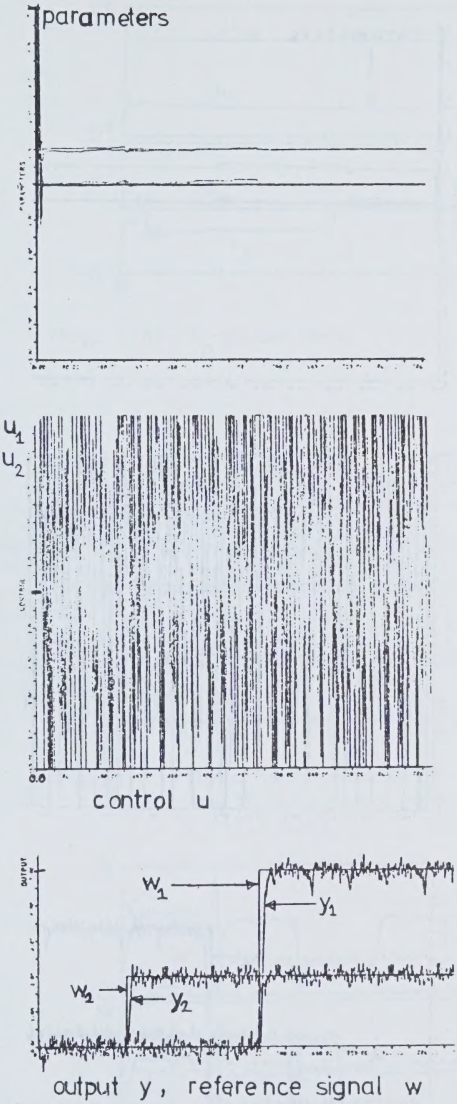


Fig. 2. Simulation with λ=0.

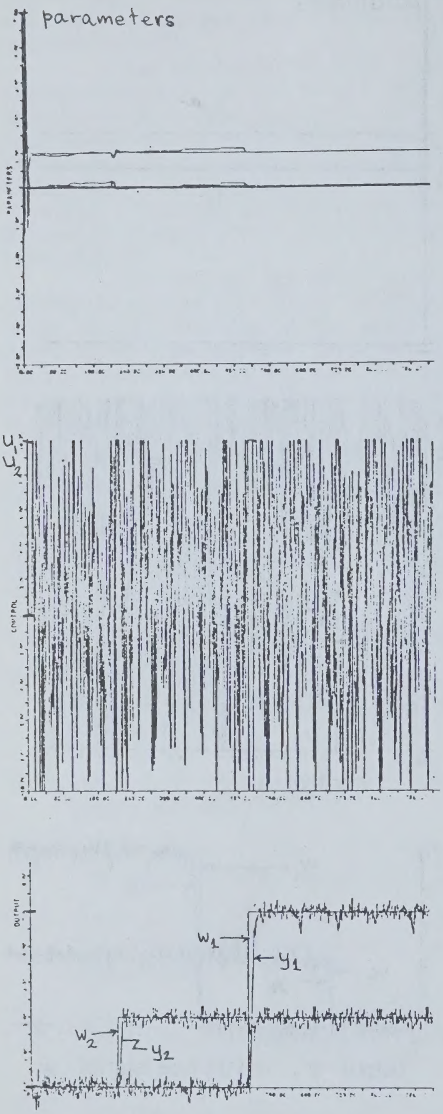


Fig. 3. Simulation with λ=0.01.

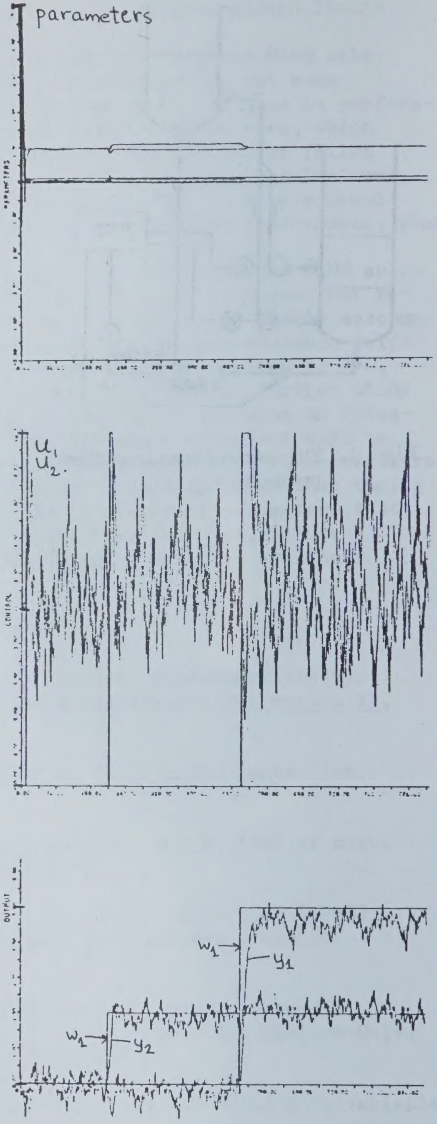


Fig. 4. Simulation with λ=0.1.

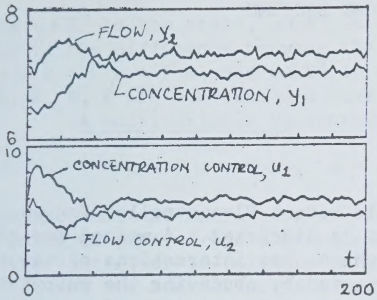


Fig. 5(a). The output response y and control u. Set point y_r=[7.2 7.2].

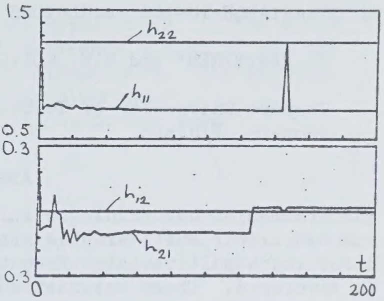


Fig. 5(d). H_o-parameters.

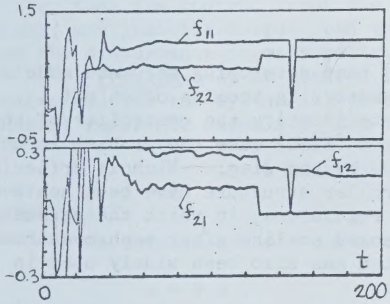


Fig. 5(b). F_o-parameters.

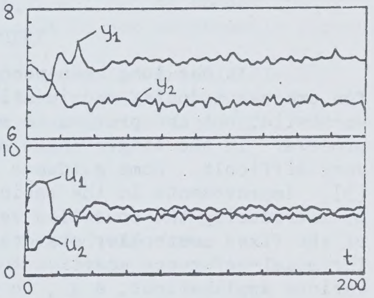


Fig. 6. The output response y and control u. Set point y_r=[7.2 6.4].

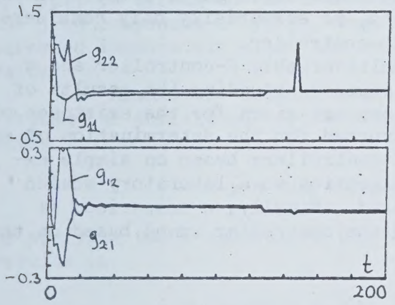


Fig. 5(c). G_o-parameters.

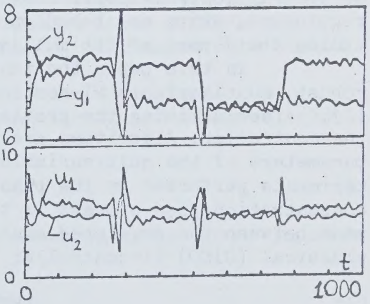


Fig. 7. The output response y and control u. Several set point changes.

MULTIVARIABLE TUNING REGULATORS FOR UNKNOWN SYSTEMS

J. PENTTINEN* and H.N. KOIVO*

Tampere University of Technology
Tampere, Finland

ABSTRACT

The problem of determining a multivariable robust PI-controller for an unknown linear multivariable stable plant is discussed. A method for constructing a multivariable P-controller, which uses interactions of the plant, is developed. These interactions are detected by observing the output of the plant subject to step-inputs. The developed P-controller form together with the robust multivariable I-controller [2] a robust multivariable PI-controller, the construction of which is also based on step-responses. The controllers are applied to a laboratory-scaled concentration-flow process. A comparison is made between the robust multivariable PI-controller and conventional PI-controller.

INTRODUCTION

It has long been recognized [8] that developing accurate models for process industry and identifying parameters in them is often not worthwile, but the problem is rather how to identify the controller of the process. In the single-input single-output (SISO) case the problem is not very difficult. Some guidance is obtained by the Ziegler-Nichols criteria [3]. Improvements in the basic PID-controller structure have been achieved by introducing the so-called self-tuning regulators, in which the parameters of the fixed controller structure are changed on-line after each measurement. The model-reference adaptive control system has also been widely used in various applications, e.g., in autopilots.

The tuning of multivariable regulators, however, has not received enough attention. Some research concerning the multivariable self-tuning regulators has been succesful [1, 5] but the theory is still incomplete. Often, in practice, fast engineering judgement is needed in on-plant tuning of several control loops. Davison [2] has constructed multivariable tuning regulators, which are based on simple tests, he essentially only considers tuning the I-part of the multivariable PID-controller.

In this paper the tuning of a multivariable P-controller and a robust, multivariable PI-controller is discussed extending the results of [2]. After defining the problem, conditions are given for the existence of the controller. Algorithms are then constructed for the determination of the parameters of the multivariable P- and PI-controllers based on simple experiments performed on the plant. An application to a laboratory-scaled concentration-flow process is then discussed. Finally, a comparison is made between the developed controller and the controller tuned based on the classical (SISO) PI-controller theory.

PRELIMINARIES

Consider the following linear time-invariant stable plant:

$$\dot{x} = A\,x + B\,u + E\,\omega,$$

$$y = C\,x + F\,\omega,$$

$$e = y_r - C\,x - F\,\omega, \qquad (1)$$

---

* This work has been supported in part by the Finnish Academy.

where $x \in E^n$ is the state, $u \in E^m$ is the control input, $e \in E^k$ is the error, which is the difference between the constant reference signal y_r and the output $y = C x + F \omega$, and $\omega \in E^q$ is a constant disturbance. The matrices A, B, C, E, F are constant and have appropriate dimensions.

A multivariable PI-controller for (1) is of the form

$$u = P e + K z \ , \quad \dot{z} = e \ , \tag{2}$$

where the constant matrices P and K are the gains of the proportional and the integral part of the controller, respectively. If $K = 0$, then $u = P e$ and the controller is called a multivariable P-controller. If $P = 0$, then $u = K z$ and the controller is a multivariable I-controller.

An important task of a multivariable PI-controller (2) in the case of a constant reference signal and a constant disturbance is to eliminate the steady-state error in the output, exactly as in the corresponding SISO-case. Another useful feature is robustness defined in [2].

As in [2], it is assumed that the order n and the plant parameters A, B, C, E, F are unknown. It is also assumed that A is asymptotically stable, that the control input can be excited, i.e., can be chosen freely enough, and that the output can be measured. It is not necessarily assumed that the disturbance is measurable.

Problem: It is desired to find a robust multivariable PI-controller (2) for the system (1) so that asymptotic regulation occurs for all constant reference signals y_r and for all constant disturbances ω.

ROBUST PI-CONTROLLER

P-controller

A multivariable P-controller for (1) is of the form

$$u = P e \ . \tag{3}$$

The idea of the proportional feedback controller is to speed up the transient. In the multivariable case, this is carried out by choosing the $m \times k$ matrix P so that the effect of the i-th component of the reference signal y_r, y_{ri}, is strong on the i-th component of the output, y, y_i, but weak on the j-th component of y, y_j, when $i \neq j$. Especially, these effects are stressed immediately after the reference signal y_r is changed into a new value.

Assume the initial state $x(0) = 0$ and the disturbance $\omega \equiv 0$. Forming the derivative of the output y, setting $t = 0$, and using eq. (1) gives

$$\dot{y} = C \dot{x} = CAx + CBu = CBu \quad \text{at} \quad t = 0 \ . \tag{4}$$

By applying the P-controller (3) to eq. (1), assuming again $x(0) = 0$, $\omega \equiv 0$, results in

$$\dot{x} = (A-BPC)x + BPy_r \ , \tag{5}$$

$$y = Cx \ , \tag{6}$$

$$y = CBPy_r \quad \text{at} \quad t = 0 \ . \tag{7}$$

Solving $x(t)$ in eq. (5) with $x(0) = 0$ and using this in eq. (6) gives

$$y(t) = - C(A-BPC)^{-1} BPy_r$$

$$+ C(A-BPC)^{-1} e^{(A-BPC)t} BPy_r \; . \tag{8}$$

Expanding the exponential term in Taylor series in eq. (8) results in

$$\begin{aligned} y &= - C(A-BPC)^{-1} BPy_r \\ &\quad + C(A-BPC)^{-1}[\sum_{k=0}^{\infty} (A-BPC)^k t^k/k!]BPy_r \\ &= CBPy_r t + C[\sum_{k=2}^{\infty} (A-BPC)^{k-1} t^k/k!]BPy_r \; . \end{aligned} \tag{9}$$

The matrix P is now chosen so that the matrix CBP is a diagonal matrix with nonzero elements on the diagonal. It can be seen in eq. (9) that this choice of P tends to reduce the effects of the components of the reference signal y_r on the components of the output y, especially when t is small.

Writing P in the form

$$P = P_o \operatorname{diag}\{p_1, p_2,\ldots,p_k\} \tag{10}$$

where P_o is a constant mxk matrix and the scalars p_i are the tuning parameters of the controller, it is enough to choose P_o as

$$P_o = (CB)^{\dagger} \tag{11}$$

where $(\cdot)^{\dagger}$ denotes the pseudoinverse of $(\cdot)$. The pseudoinverse [7] of the matrix CB is given by

$$(CB)^{\dagger} = B^T C^T (CBB^T C^T)^{-1} \tag{12}$$

if the rank of CB satisfies

$$\operatorname{rank}(CB) = k \; . \tag{13}$$

So the P-controller (3) is of the form

$$u = (CB)^{\dagger} \operatorname{diag}\{p_1, p_2,\ldots,p_k\} \tag{14}$$

if $(CB)^{\dagger}$ exists.

The following algorithm gives an experimental way for determining the matrix CB. It is assumed that the disturbance $\omega \equiv 0$ and that the initial condition $x(0) = 0$.

Algorithm I.

Step 1. Let $u(t) \equiv 0$. When the steady-state ($y(\infty) = 0$) is achieved, choose a constant input $u(t) \equiv u_1$ ($\neq 0$) and apply it to the plant. Then $\dot{y}(0) = \dot{y}_1 = CBu_1$. From the corresponding step-response $\dot{y}_1$ can be computed, e.g., graphically.

Step 2. Repeat step 1 with a constant control u_2, which is linearly independent of u_1. Then $\dot{y}(0) = \dot{y}_2 = CBu_2$. Compute $\dot{y}_2$ from the corresponding step-response.

Step m. Repeat step 1 with a constant control u_m, which is linearly independent of $u_2,\ldots,u_m$. Then $\dot{y}(0) = \dot{y}_m = CBu_m$. Compute $\dot{y}_m$ from the corresponding step-response.

As a result

$$CB = [\dot{y}_1 \; \dot{y}_2 \ldots \dot{y}_m][u_1 \; u_2 \ldots u_m]^{-1} \; . \tag{15}$$

The existence of the P-controller (14) can now be checked by conditions (15) and (13). If the P-controller exists, the tuning parameters can be found experimentally.

Remark: Note that the closed-loop system (5) remains stable at least with small p_i's.

Robust I-controller

The I-part of the controller in the step-case is given in [2], where the matrix K is defined as

$$K = \varepsilon(T)^{\dagger} = \varepsilon(-CA^{-1}B)^{\dagger} \tag{17}$$

where T is called the tracking steady-state gain parameter of the system (1)-(3). A scalar tuning parameter ε can always be found so that the closed-loop system is stable.

The next algorithm of [2] provides a way to determine experimentally the parameter T. Again, assume the disturbance $\omega \equiv 0$.

Algorithm II.

Step 1. Apply a constant input $u(t) \equiv u_1$, $u_1 \neq 0$, to the open-loop plant. The system (1) is stable, so a steady-state solution $y(\infty) = y_1$, exists. Measure y_1.

Step 2. Repeat step 1 with a constant input u_2, which is linearly independent of the previous input u_1. Measure y_2.

Step m. Repeat step 1 with a constant input u_m, which is linearly independent of the previous inputs $u_1, \ldots, u_{m-1}$. Measure y_m.

As a result

$$T = [y_1\ y_2\ \ldots\ y_m][u_1\ u_2\ \ldots\ u_m]^{-1} \tag{18}$$

Davison's necessary and sufficient condition for the existence of the robust I-controller which eliminates the steady-state error in the output for all constant disturbances is (see Th. 2. [2])

$$\operatorname{rank}(T) = k\ . \tag{19}$$

Robust PI-controller

The robust multivariable PI-controller can now be constructed (if it exists) based on Algorithms I and II in the following way:

Determine the gain matrices P_o and K (if they exist) of the P- and I-controller, respectively, using Algorithms I and II. The step-inputs needed can be the same for both Algorithms. Then tune the P-controller for the plant by adjusting the tuning parameters $p_1, \ldots, p_k$, (starting with small positive values) so that the closed-loop response (for a step input reference signal) is as good as possible. Make these parameters then little smaller and tune the PI-controller by adjusting the tuning parameter of the I-controller, ε, starting with small positive values so that the closed-loop (PI-controlled) response (for a step input reference signal) has the maximum speed of response.

Discussion of Previous Results: The previous sections have described a procedure by which a robust multivariable PI-controller for an unknown stable linear time-invariant plant can be found by using simple experiments. The advantage of the I-part of the controller is that it makes the controller robust as is shown in [2]. The utilization of the I-controller alone, however, does not achieve a rapid, corrective response for deviations in the output error. Davison's suggestion in [2] for a quicker response is: "Speed up the dynamics of the plant as much as possible using heuristic methods, e.g., by using the output control $u = K_1 y + K_2 \dot{y}$. ... Find a robust feedback controller for the resultant controlled system". Here, a simple, systematic procedure for the determination of the multivariable P-controller has been given. The advantage of this P-controller is that it can efficiently use interactions existing between the control inputs and the outputs of the system after the reference signal and/or the disturbance are changed. So the P-controller forces the output to the right direction after changes in the reference and/or in the disturbance, and thus achieves a rapid corrective response. The robust multivariable PI-controller has the properties of both the P-controller and the robust I-controller, and thus is an efficient multivariable controller.

In eq. (17) there is only one tuning parameter whereas in (10) there are k tuning parameters. It would, of course, be possible to use also k tuning parameters in (17) and improve regulation. On the other hand, it is possible to use one tuning parameter with both controllers. This simplifies the tuning of the PI-controller.

The multivariable PI-controller structure (2) with P and K as in (10) and in (17), respectively, is the same as the controller structure proposed by Rosenbrock [10] and further discussed by MacFarlane and Kouvaritakis [6]. These authors use frequency domain approach, and the square transfer function matrix G(s) of the system (1) is assumed to be known. The calculation of the PI-controller gain matrices is based on the steady-state information (s = 0) of the transfer-functions matrix and the information at high frequencies. The tuning of the controller must be carried out experimentally. Here, the tuning method is quite simple, no model is constructed, and one needs the steady-state information and the derivatives of the step-responses which are easily evaluated graphically.

APPLICATION

The algorithms introduced in the previous section for construction P-, I-, and PI-controllers are applied to a laboratory-scaled concentration -flow process described in companion paper [5]. In addition, a classical SISO PI-controller is applied, and the results are compared.

Let

$$\begin{aligned} u &= [\tilde{u}_1 \; \tilde{u}_2]^T - [\tilde{u}_{01} \; \tilde{u}_{02}]^T, \\ y &= [\tilde{y}_1 \; \tilde{y}_2]^T - [\tilde{y}_{01} \; \tilde{y}_{02}]^T, \\ y_r &= [\tilde{y}_{r1} \; \tilde{y}_{r2}]^T - [\tilde{y}_{01} \; \tilde{y}_{02}]^T \end{aligned} \qquad (20)$$

denote the control input, the output, and the reference signal of the process, respectively, so that $\tilde{u}_1$ is the control voltage of the flow pump, $\tilde{y}_1$ is the measurement voltage of the flow, $\tilde{y}_{r1}$ is the reference voltage of the flow, and $\tilde{u}_{01}$, $\tilde{y}_{01}$ are the corresponding operating point values, and $\tilde{u}_2$, $\tilde{y}_2$, $\tilde{y}_{r2}$ are the corresponding values of the concentration.

The process can be regarded as a linear time-invariant two input-two output process around the operating point. The operating point for the control input is chosen to be

$$\tilde{u}_o = [5, 5]^T , \qquad (21)$$

which causes the operating point for the output to be

$$\tilde{y}_o = [7.3, \; 6.5]^T. \qquad (22)$$

Two cases are examined with each controller. In the first case, the output

$\tilde{y}$ is regulated from

$$y = [7.8 \quad 6.2]^T \tag{23}$$

to

$$\tilde{y}_r = [7.8 \quad 7.4]^T \ . \tag{24}$$

This situation gives a possibility to consider interactions between the outputs when only one (concentration) of them is changed. In the second case, the output $\tilde{y}$ is regulated from

$$\tilde{y} = [7.8 \quad 6.2]^T \tag{25}$$

to

$$\tilde{y}_r = [6.6 \quad 7.4]^T. \tag{26}$$

This situation illustrates the performance of a controller.

Multivariable P-controller

For the multivariable P-controller the first task is to determine the matrix CB. This is done by using Algorithm I. The constant inputs are chosen as

$$u_1 = [1 \ 0], \quad u_2 = [0 \ 1] \ . \tag{27}$$

The corresponding outputs are displayed in Fig's 1 and 2. The derivatives of the outputs are computed from the Fig's as

$$\begin{aligned} \dot{y}_1 &= [0.010 \quad -0.013]^T \\ \dot{y}_2 &= [0.015 \quad 0.014]^T \end{aligned} \tag{28}$$

which results in

$$CB = \begin{bmatrix} 0.010 & 0.015 \\ -0.013 & 0.014 \end{bmatrix} \tag{29}$$

and

$$P_o = (CB)^{\dagger} = \begin{bmatrix} 41.79 & -44.78 \\ 38.81 & 29.85 \end{bmatrix}. \tag{30}$$

So the P-matrix is of the form

$$P = \begin{bmatrix} 41.79 & -44.78 \\ 38.81 & 29.85 \end{bmatrix} \begin{bmatrix} p_1 & 0 \\ 0 & p_2 \end{bmatrix} \tag{31}$$

where p_1 and p_2 are tuning parameters. Satisfactory values for p_1 and p_2 where found to be:

$$p_1 = 0.04 \quad p_2 = 0.06 \ . \tag{32}$$

Fig's 3 and 4 represent the two different situations for the multivariable

P-controller (31)-(32). Fig. 3 shows that this controller is able to exploit both control inputs although only one of the outputs is to be changed. Fig. 4 displays the situation where both outputs are changed.

Multivariable I-controller

Next the multivariable I-controller of [2] is examined. The experiments carried out in connection with the multivariable P-controller can be used for determining the K-matrix according to Algorithm II. In Fig's 1 and 2 the steady-state values of the outputs corresponding to the inputs (27) can be evaluated:

$$y_1 = [0.6 \quad -0.7]^T \qquad y_2 = [0.7 \quad 0.6]^T . \tag{33}$$

Utilizing (27), (33) and (18) gives

$$K = \varepsilon \begin{bmatrix} 0.71 & -0.82 \\ 0.82 & 0.71 \end{bmatrix} \tag{34}$$

A satisfactory value for ε was found to be $\varepsilon = 0.1$. The resulting I-controllers are displayed in Fig's 5 and 6. Clearly, the multivariable I-controller eliminates the steady-state error as is claimed in [2], but it is obvious that it is not fast enough to be used by itself.

Multivariable PI-controller

The parameters of the multivariable PI-controller are given in eq's (30) and (34). The tuning parameters in the P-controller are chosen to be the same as above in eq. (32). In this manner, the influence of the I part is best seen and the results easily compared. A satisfactory value for the tuning parameter of the I-controller, ε, was found to be $\varepsilon = 0.05$. Fig's 7 and 8 show the results of the regulation of the multivariable PI-controller. The elimination of the steady-state error is apparent. The response is much faster than in the case of the multivariable I-controller.

Classical PI-controller

If the classical SISO PI-controller is used, then the matrices P and K in (2) are diagonal matrices, i.e., the interactions are neglected. The tuning of this controller was carried out by first applying Ziegler-Nichols adjustment rules [3] and then experimentally improving the (diagonal) values of P and K. Satisfactory values were found to be

$$P = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix} \tag{35}$$

$$K = \begin{bmatrix} 0.05 & 0 \\ 0 & 0.02 \end{bmatrix} . \tag{36}$$

The results are shown in Fig's 9 and 10. The purpose was to change only output y_2, but as seen in Fig. 9 the output y_1 is also influenced indicating considerable interaction. Comparing results in Fig's 7 and 9, it is obvious that the proposed PI-controller (31)-(34) is superior to the classical PI-controller (35)-(36). Comparing then the results between the two PI-controllers in the case where both outputs are to be changed, Fig's 8 and 10 indicate the proposed controller to be more efficient; it uses smaller inputs and has shorter rise time. The final elimination of the steady-state error is achieved within about the same time by each controller.

CONCLUSION

A new method for tuning a multivariable PI-controller has been proposed extending the I-controller of [2]. The model of the plant is not known, only that the system is linear time-invariant and stable. The tuning procedure is simple and the associated experiments are easily performed. The experiments resemble those carried out on plant in practical, multiloop tuning of the SISO PID-controllers. The multivariable PI-controller exhibits similar properties as the corresponding scalar controller: it has good transient bahavior, it is robust, and it tracks asymptotically with zero steady-state error.

A laboratory-scaled concentration-flow process was used to test the proposed multivariable PI-controller and to compare its bahavior with other similar controllers. Initial tests have been quite promising indicating the tuned multivariable PI-controller to be superior to the tuned multivariable I-controller of the tuned SISO PI-controller.

REFERENCES

[1] Borisson U. (1975). Self-tuning regulators- Industrial application and multivariable theory. Report 7513. Dept. of Automatic Control, Lund Inst. of Tech., Lund, Sweden.

[2] Davison E.J. (1976). Multivariable tuning regulators: The feedforward and robust control of a general servomechanism problem. IEEE Trans. Autom. Control, AC-21, 35-47.

[3] Harriott H.P. Process Control. McGraw-Hill, New York.

[4] Keviczky L., Hetthéssy J., Hilger M., and J. Kolostori. (1978). Self-tuning adaptive control of cement raw material blending. Automatica 14, 525-534.

[5] Koivo H.N., Tanttu J. and J. Penttinen. (1979). Experimental evaluation of multivariable self-tuning controllers. This conference.

[6] MacFarlane A.G.J. and B. Kouvaritakis. (1977). A design technique for linear multivariable feedback systems. Int. J. Control 25, 837-874.

[7] Penrose R. (1955). A generalized inverse for matrices. Proc. Camdridge Philos. Soc. 51, 406-413.

[8] Peterka V. and K.J. Åström (1973). Control of multivariable systems with unknown but constant parameters. IFAC Symposium on Identification and System Parameter Estimation. The Hague, Netherlands, 535-544.

[9] Rajala M. (1976). A pilot plant for concentration control. Sähkö-Electricity in Finland, 10, 385-389.

[10] Rosenbrock H. (1974). Computer-aided control system design. Academic-Press. New York.

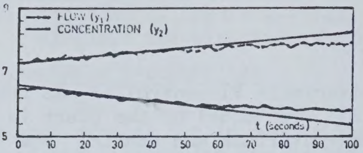


Fig. 1. The step-input responses of the open plant. The control $u(t) = [1 \ \ 0]^T$ is applied from t = 0.

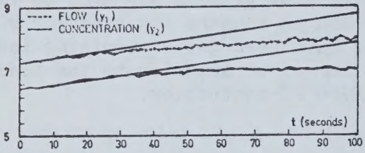


Fig. 2. The step-input responses of the open plant. The control $u(t) = [0 \ \ 1]^T$ is applied from t = 0.

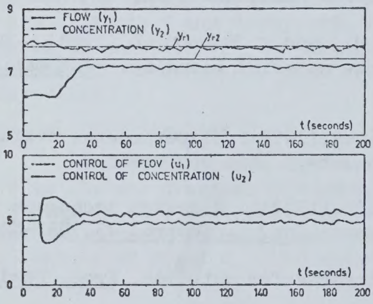


Fig. 3. The output-responses and the control inputs in the P-controller case. The reference signal $\tilde{y}_r$ is changed to $\tilde{y}_r = [7.8 \ \ 7.4]$ at t = 10 s.

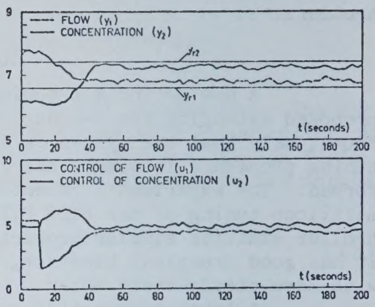


Fig. 4. The output responses and the control inputs in the P-controller case. The reference signal $\tilde{y}_r$ is changed to $\tilde{y}_r = [6.6 \ \ 7.4]$ at t = 10 s.

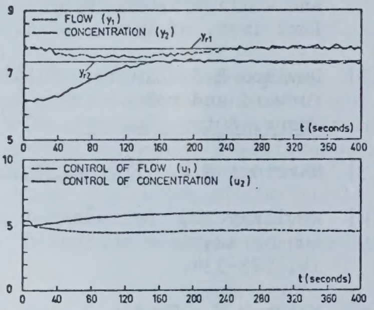


Fig. 5. The output responses and the control inputs in the I-controller case. The reference signal $\tilde{y}_r$ is changed to $\tilde{y}_r = [7.8 \ \ 7.4]$ at t = 10 s. Note the time-scale.

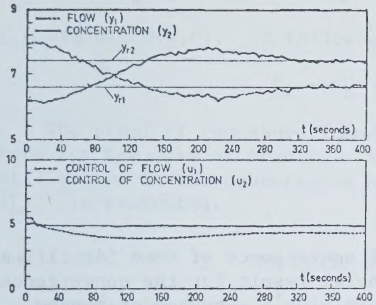


Fig. 6. The output responses and the control inputs in the I-controller case. The reference signal $\tilde{y}_r$ is changed to $\tilde{y}_r$ = [6.6 7.4] at t = 10 s. Note the time-scale.

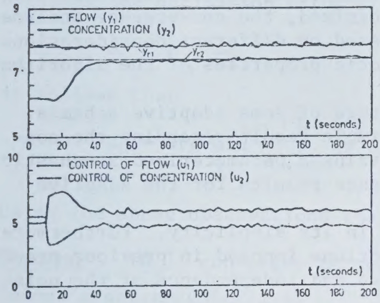


Fig. 7. The output responses and the control inputs in the multivariable PI-controller case. The reference signal $\tilde{y}_r$ is changed to $\tilde{y}_r$ = [7.8 7.4] at t = 10 s.

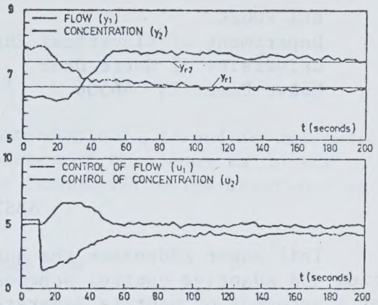


Fig. 8. The output responses and the control inputs in the PI-controller case. The reference signal $\tilde{y}_r$ is changed to $\tilde{y}_r$ = [6.6 7.4] at the t = 10 s.

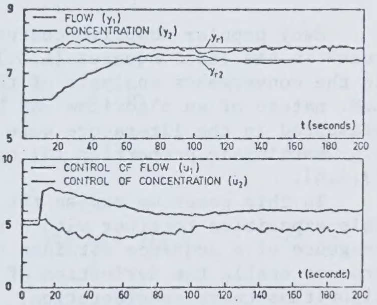


Fig. 9. The output responses and the control inputs in the classical PI-controller case. The reference signal $\tilde{y}_r$ is changed to $\tilde{y}_r$ = [7.8 7.4] at t =10 s.

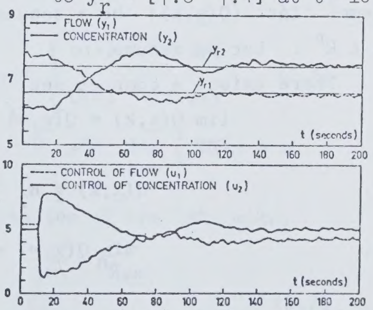


Fig. 10. The classical PI-controller. The reference signal $\tilde{y}_r$ is changed to $\tilde{y}_r$=[6.6 7.4] at t=10s.

ON STRONG CONSISTENCY OF SOME ADAPTIVE ALGORITHMS

ELI FOGEL
Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556

ABSTRACT

This paper addresses the question of convergence of some identification and adaptive control schemes. A general result for the convergence of a sequence of vectors generated by minimizing a sequence of random variables parameterized by these vectors is obtained.

This result is applied to the least squares identification scheme and to Landau's recursive algorithms. Convergence w.p.1 and in the mean^2 is demonstrated under milder conditions than usually are assumed in the literature.

1. INTRODUCTION

Many popular adaptive control and estimation procedures have their roots in the least squares (L.S.) method. This fact is often overlooked in the convergence analysis of these methods [1-8]. Moreover, even if the L.S. nature of an algorithm has been recognized, the convergence analyses presented in the literature were often based on different considerations, e.g. martingale properties [5] or asymptotic properties of the algorithm [3,4,6].

In this paper we expose the L.S. nature of some adaptive schemes. This exposition together with a rather simple result regarding the convergence of a sequence obtained by minimizing a parameterized stochastic process enable the derivation of convergence results for the adaptive algorithms under consideration.

The significance of our approach is in its simplicity. Furthermore it enables the relaxation of some restrictions imposed in previous proofs e.g., stationarity of the measured process and independence of the noise sequence.

The following lemma is the basis of our analysis.

Lemma Let $\{Q(\underline{x},k)\}$ be a sequence of random variables parameterized by $\underline{x} \in \mathcal{R}^n$. Let us assume

1. There exists a compact set $\mathcal{X} \in \mathcal{R}^n$ such that

$$\lim_{k\to\infty} Q(\underline{x},k) = Q(\underline{x},\infty) \le B_1 < \infty \quad \forall\, \underline{x} \in \mathcal{X} \quad \text{w.p.1} \qquad (1)$$

2.

$$Q(\underline{x},k) \ge B_2 > -\infty \quad \forall\, \underline{x} \in \mathcal{R}^n \quad \text{w.p.1} \qquad (2)$$

3.

$$\min_{\underline{x}\in\mathcal{R}^n} Q(\underline{x},\infty) = \min_{\underline{x}\in\mathcal{X}} Q(\underline{x},\infty) \quad \text{w.p.1} \qquad (3)$$

Using the notation $\underline{\Gamma}_k \triangleq \arg\min_{\underline{x}} Q(\underline{x},k)$ to denote

$$Q(\underline{\Gamma}_k k) \leq Q(\underline{x},k) \quad \forall\ \underline{x} \in R^n \quad \text{and similarly define}$$

$\underline{\Gamma} \triangleq \arg\min\limits_{\underline{x}} Q(\underline{x},\infty)$, it follows that

$$\underline{\Gamma}_k \longrightarrow \underline{\Gamma} \qquad \text{w.p.1} \tag{4}$$

The proof of the above lemma is rather simple [10] and is omitted here for the sake of brevity. The importance of the lemma is in its utilization in the convergence proofs for random variables generated as $\{\underline{\Gamma}_k\}$ is generated.

2. THE LEAST SQUARE ALGORITHM

Consider the n-th order AR model

$$z_k = \underline{a}^T \underline{Z}_k + \omega_k \tag{2.1}$$

where

$$\underline{a} \in R^n, \quad \underline{Z}_k^T = [z_{k-1}, \ldots z_{k-n}] \quad \text{and} \quad \{\omega_k\} \ \text{is a zero mean}$$

white noise sequence. The L.S. estimate of $\underline{a}$ is given by

$$\hat{\underline{a}} = \arg\min_{\underline{x}} \frac{1}{k} \sum_{i=1} (\underline{x}^T \underline{Z}_i - z_i)^2 \tag{2.2}$$

Defining the estimation error at time instant k by

$$\tilde{\underline{a}}_k \triangleq \underline{a} - \hat{\underline{a}}_k \tag{2.3}$$

it follows that

$$\tilde{\underline{a}}_k = \arg\min_{\underline{x}} \frac{1}{k} \sum_{i=1}^{k} (\underline{x}^T \underline{Z}_i - \omega_i)^2 = \arg\min_{\underline{x}} Q(\underline{x},k) \tag{2.4}$$

Using the above observations the following result is obtained:

Theorem 1. $\{\tilde{\underline{a}}_k\}$ converges to zero w.p.1 if $\{\omega_i\}$ is white and second order ergodic, i.e.

$$E(\omega_i) = 0 \quad ; \quad E[\omega_i \omega_j] = Q\ \delta_{ij} \tag{2.5}$$

$$\sum_{i=1}^{k} (\omega_i) = o(k) \qquad\qquad \sum_{i=1}^{k} \omega_i \omega_j = \begin{cases} o(k) & \text{if } i \neq j \\ O(k) & \text{if } i = j \end{cases} \tag{2.6}$$

where $o(k)$ and $O(k)$ mean $\lim\limits_{k\to\infty} \frac{1}{k}\, o(k) = 0$ and $\lim\limits_{k\to\infty} \frac{1}{k}\, O(k) > o$

Proof. Consider the state space representation of the AR model of eq. (2.1):

$$\underline{Z}_{k+1} = A\underline{Z}_k + B\ \omega_k \tag{2.7}$$

Substituting (2.7) into $Q(\underline{x},k)$ yields

$$
\begin{aligned}
k\cdot Q(\underline{x},k) &= \sum_{i=1}^{k}\Big(\underline{x}^T\sum_{j=1}^{i}A^{i-j}B\ \omega_{j-1}-\omega_i\Big)^2 \\
&= \sum_{i=1}^{k}\Big[\underline{x}^T\sum_{j=1}^{i}\ \sum_{m=1}^{i}A^{i-j}B(A^{i-m}\underline{B})^T\omega_{j-1},\omega_{m-1}\underline{x}-2\underline{x}^T\sum_{j=1}^{i}A^{i-j}B\omega_{j-1}\omega_i+\omega_i^2\Big] \\
&= \underline{x}^T\Big[\sum_{j=1}^{k}\ \sum_{m=1}^{k}(A^{j-1}\underline{B})(A^{m-1}B)^T\sum_{i=\max(j,m)}^{k}\omega_{i-j},\omega_{i-m}\Big]\underline{x} \\
&\quad - 2\underline{x}^T\sum_{j=1}^{k}A^{j-1}B\sum_{i=j}^{k}\omega_{i-j}\ \omega_i+\sum_{i=1}^{k}\omega_i^2 \\
&= \underline{x}^T\Big[\sum_{j=1}^{k}(A^{j-1}B)(A^{j-1}B)^T O(k)+\sum_{j=1}^{k}\ \sum_{\substack{m=1\\ m\neq j}}^{k}(A^{j-1}B)(A^{m-1}B)^T o(k)\Big]\underline{x} \\
&\quad - 2\underline{x}^T\sum_{j=1}^{k}A^{j-1}B\ o(k)+\sum_{i=1}^{k}\omega_i^2\Big] \\
&= \underline{x}^T\Big[\sum_{j=1}^{k}(A^{j-1}B)(A^{j-1}B)^T O(k)\Big]\underline{x}+\Big(\sum_{j=1}^{k}\underline{x}^TA^{j-1}\underline{B}-1\Big)^2 o(k) \\
&\quad - \underline{x}^T\sum_{j=1}^{k}(A^{j-1}B)(A^{j-1}B)^T\underline{x}\ o(k)+\sum_{i=1}^{k}\omega_i^2 \\
&= \underline{x}^T\sum_{j=1}^{k}(A^{j-1}B)(A^{j-1}B)^T O(k)\underline{x}+\Big(\sum_{j=1}^{k}x^TA^{j-1}B-1\Big)^2 o(k)+\sum_{i=1}^{k}\omega_i^2
\end{aligned}
\tag{2.8}
$$

In the above manipulations the identities $O(k) + o(k) = O(k)$ and $o(k) + o(k) = o(k)$ were used.

Let X be the set on which $Q(\underline{x},k)$ is bounded $\forall\, k$.

Obviously X is not an empty set since $Q(\underline{o},k)$ is bounded.

On X we have:

$$\arg\min_{\underline{x}} Q(\underline{x},\infty) = \underline{o} \tag{2.9}$$

Thus it follows from the main lemma that $\underline{\tilde{a}}_k$ converges to zero w.p.1 .

Remarks. (1) In the sequential version of the L.S. algorithm, the error at time instant k consists of $\underline{\tilde{a}}_k$ as defined in (2.4) and an additional term: $P_k\,\underline{\tilde{a}}_o$ where $\underline{\tilde{a}}_o = \underline{a} - \underline{\hat{a}}_o$ and $P_k = \left(\sum_{i=1}^{k}\underline{z}_i\underline{z}_i^T + P_o^{-1}\right)^{-1}$.

It can be shown that w.p.1 $\underline{P}_k \to o$ [10] and thus the convergence proof for the sequential L.S. is established.

(2) Earlier attempts to prove strong convergence of the L.S. algorithm were based on the sequential version and resorted to complex statistical arguments [4-6, 11-13]. Furthermore, most of the results were based on i.i.d assumption regarding the noise sequence [1-6] or were restrictive in terms of the model spectrum [1-4, 6]. The restriction on $\{\omega_i\}$ in Theorem 1 is satisfied under rather simple conditions as stated below.

Corollary 1. The L.S. algorithm converges w.p.1 if $\{\omega_i\}$ is weakly stationary of order 4 and its 4-th moment satisfy

$$E[\omega_i \ \omega_j \ \omega_{i-m} \ \omega_{j-k}] = \begin{cases} (Q-R^2)\lambda_1^P + R^2 & \text{if } i=j \ m=k \\ H \ \lambda_2^P & \text{otherwise} \end{cases}$$

where $|\lambda_1|<1$, $|\lambda_2|<1$, $o \le k \le j$, $o \le m \le i$ and

$$p = \max(i-j+k \ , j-i+m) .$$

Proof. See [10].

3. LANDAU'S ALGORITHMS [6-8].

Landau has presented two procedures for the identification of the ARMA model

$$z_k = \underline{a}^T \underline{Z}_k + \underline{b}^T \underline{U}_k + \omega_k = \underline{\theta}^T \underline{x}_k + \omega_k \tag{3.1}$$

where $\underline{Z}_k$ is defined as earlier, $\underline{U}_k^T \triangleq (u_k, u_{k-1}, \ldots u_{k-m}), \{u_i\}$ is measurable,

$\underline{\theta}^T \triangleq [\underline{a}^T, \underline{b}^T]$ and $\underline{x}_k^T \triangleq [\underline{Z}_k^T, \underline{U}_k^T]$.

The identification procedures of $\underline{\theta}$ are presented and analyzed below.

Algorithm 1 [7].

$$\underline{\hat{\theta}}_k = \underline{\hat{\theta}}_{k-1} + P_k \, \underline{\hat{x}}_k \, \bar{\varepsilon}_k \tag{3.2}$$

$$P_k^{-1} = \sum_{i=1}^{k} \underline{\hat{x}}_i \, \underline{\hat{x}}_i^T + P_o^{-1} \tag{3.3}$$

$$\underline{\hat{x}}_k^T = [\underline{\hat{Z}}_k^T, \underline{U}_k^T] \ ; \ \hat{z}_k = \underline{\hat{\theta}}_k^T \, \underline{\hat{x}}_k \tag{3.4}$$

and

$$\varepsilon_k = z_k - \hat{z}_k \ ; \ \bar{\varepsilon}_k = z_k - \underline{\hat{\theta}}_{k-1}^T \underline{\hat{x}}_k + \sum_{i=1}^{k} c_i \ \varepsilon_{k-i} \tag{3.5}$$

Landau has presented a deterministic convergence proof (i.e., $\omega_k = o$) for his algorithm [7]. Ljung has presented a proof of convergence w.p.1 for the stochastic case, assuming i.i.d noise sequence and stationary model [6]. The similarity of the algorithm to the L.S. algorithm suggests that these are closely related algorithms as stated below:

Proposition 1. Define

$$\bar{\varepsilon}_i(\hat{\theta}) \triangleq z_i - \underline{\hat{\theta}}^T \underline{\hat{x}}_i + \sum_{j=1} c_j \varepsilon_{i-j} \tag{3.6}$$

then $\underline{\hat{\theta}}_k$ of (3.2) satisfies

$$\underline{\hat{\theta}}_k = \arg \min_{\hat{\theta}} J_k(\underline{\hat{\theta}}) + P_k \underline{\hat{\theta}}_o \tag{3.7}$$

where

$$J_k(\underline{\hat{\theta}}) = \sum_{i=1}^{k} (\bar{\varepsilon}_i(\underline{\hat{\theta}}))^2 \tag{3.8}$$

The proof of the proposition is straightforward [10]. It follows from the above that the estimation error can be written as

$$\underline{\tilde{\theta}}_k \triangleq \underline{\theta} - \underline{\hat{\theta}}_k = \underline{\tilde{\theta}}^u_k + \underline{\tilde{\theta}}^\omega_k \tag{3.9}$$

where $\underline{\tilde{\theta}}^u_k$ is the deterministic error, i.e., the error in the estimate in absence of noise and where $\underline{\tilde{\theta}}^\omega_k$ is the error due to the noise. Furthermore, it can be shown that

Proposition 2.

$$\underline{\tilde{\theta}}^\omega_k = \arg \min_{\underline{\tilde{\theta}}} \sum_{j=1}^{k} [\underline{\tilde{\theta}}^T \underline{\hat{x}}_j - \underline{V}_j^T(\underline{a} + \underline{C})]^2 \tag{3.10}$$

where $\underline{V}_j$ is the state vector of the identified model for $\{u_i = o\}$ and $\underline{C}^T = (c_1, c_2, \dots c_n)$ c_i, i=1, ... n are defined in (3.5).

Theorem 2. If $\sum_{i=o}^{n} c_i q^i / (1 - \sum_{i=1}^{n} a_i q^i)$ is positive real and $\{\omega_i\}$ satisfies the conditions of Theorem 1 (Corollary 1) then $\underline{\tilde{\theta}}_k$ converges to o w.p.1 .

Proof. The positive real assumption guarentees the convergence of $\tilde{\theta}^u_k$ as proved by Landau [7]. The convergence proof for $\tilde{\theta}^\omega_k$ follows the steps of the convergence proof of Theorem 1.

Remarks. A key condition in Landau's and Ljung's proofs is the positive real assumption. We impose this condition only to guarentee the convergence of $\underline{\tilde{\theta}}^u_k$. This assumption does not play any role in the convergence of $\underline{\tilde{\theta}}^\omega_k$. Landau relaxed the positive real assumption [8] by allowing the adaptive adjustment of the vector $\underline{C}$ as in the following algorithm:

Algorithm 2 [8]. The same as Algorithm 1 with the following modifications:

$$\underline{\hat{x}}_k \text{ is replaced by } \underline{\hat{x}}^e_k \triangleq (\underline{\hat{x}}^T_k, \underline{Z}^T_k - \underline{\hat{Z}}^T_k)^T$$

$$\underline{\hat{\theta}}_k \text{ is replaced by } \underline{\hat{\theta}}^e_k \triangleq (\underline{\hat{\theta}}^T_k, -\underline{\hat{C}}^T_k)^T$$

and $\underline{C}$ is replaced by $\underline{\hat{C}}_k$ to yield the new algorithm

$$\underline{\hat{\theta}}^e_k = \underline{\hat{\theta}}^e_{k-1} + \underline{P}_k \underline{x}^e_k \bar{\varepsilon}_k \tag{3.11}$$

$$\bar{\varepsilon}_k = z_k - (\underline{\hat{\theta}}^e_{k-1})^T \underline{\hat{x}}^e_k \tag{3.12}$$

and

$$P^{-1}_k = \sum_{i=1}^{k} (\underline{\hat{x}}^e_i)(\underline{\hat{x}}^e_i)^T + P^{-1}_o \tag{3.13}$$

Defining the error vector

$$\underline{\tilde{\theta}}^e_k \triangleq (\theta^T - \hat{\theta}^T_k, \underline{a}^T + \underline{\hat{C}}^T_k)^T \tag{3.14}$$

The following theorem follows:

Theorem 3

$$\underline{\hat{\theta}}^e_k = \arg\min_{\underline{\hat{\theta}}^e} \left[\sum_{i=1}^{k} (z_i - (\underline{\hat{x}}^e_i)^T \underline{\hat{\theta}}^e)^2 \right] + P_k \underline{\hat{\theta}}^e_o \tag{3.15}$$

$$\underline{\tilde{\theta}}_k = \arg\min_{\underline{\tilde{\theta}}} \left[\sum_{i=1}^{k} ((\underline{\tilde{\theta}}^e)^T \underline{\hat{x}}^e_i + \omega_i)^2 \right] + P_k \underline{\tilde{\theta}}^e_o \tag{3.16}$$

$\underline{\tilde{\theta}}_k$ converges to zero w.p.1 if $\{\omega_i\}$ satisfies the constraints of Theorem 1 (Corollary 1).

Proof. (3.15)(3.16) follow from the definitions and the convergence statement can be proved as Theorem 1 is proved.

Remarks. (1) It is not claimed here that $\underline{\hat{C}}_k \to -\underline{a}$. Although such convergence may occur, it is also possible that $(\underline{\hat{C}}_k + \underline{a})$ converge to a constant while $\frac{1}{k} \sum^{k} \underline{\tilde{Z}}_i \underline{\tilde{Z}}^T_i \to o$.

(2) The above proof is independent of the hyperstability analysis of [8]. Note that stochastic convergence proof for Algorithm 2 is presented here for the first time.

4. MEAN SQUARE CONVERGENCE

The statements of the previous section can be easily extended to convergence in the mean square sense. We present the least square convergence corresponding to Theorem 1. The proof of least square convergence corresponding to Theorems 2 and 3 follow the same arguments.

Theorem 4. Under the conditions of Theorem 1, the L.S. algorithm converges in the m.s. sense.

Proof. By Burrill, Theorem 9-5G [9] $\{g_i\}$ converges to g in mean^P if it converges in probability and $\{|g_i|^P\}$ is uniformly integrable. By the definition of $\{\tilde{\underline{a}}_k\}$ we have

$$\left|\frac{1}{k}\sum_{i=1}^{k}(\tilde{\underline{a}}_k^T Z_i - \omega_i)^2\right| \leq \frac{1}{k}\sum_{i=1}^{k}\omega_i^2$$

Using arguments as in (2.8) we have

$$\tilde{\underline{a}}_k^T\left(\frac{1}{k}\sum_{i=1}^{k}\underline{Z}_i\,\underline{Z}_i^T\right)\tilde{\underline{a}}_k \leq \frac{2}{k}\sum_{i=1}^{k}\omega_i^2$$

to yield

$$||\tilde{\underline{a}}_k||^2 \leq \frac{2}{k}\sum_{i=1}^{k}\omega_i^2 \Big/ \lambda_{min}\left[\frac{1}{k}\sum_{i=1}^{k} Z_i\, Z_i^T\right]$$

It can be shown [10] that the denominator does not converge to zero and thus $||\tilde{\underline{a}}_k||^2$ is bounded, thus uniformly integrable to complete the proof.

5. CONCLUSIONS

A fundamental lemma regarding the convergence of a sequence of random variables obtained by minimizing a parameterized stochastic sequence. The lemma is instrumental in the convergence analysis of the L.S. algorithms. Strong convergence of the method is established under no restriction on the model spectrum and under mild restrictions on the noise.

Furthermore, the L.S. nature of other algorithms [7,8] is recognized to yield simple convergence results for these algorithms.

Finally we note that m.s. convergence can be easily tied to convergence w.p.1 for L.S. procedures.

ACKNOWLEDGEMENT

The support of the Department of the Navy, Office of Naval Research, grant no. N00014-78-C-0444 is gratefully acknowledged.

REFERENCES

[1] H.B. Mann and A. Wald, "On the Statistical Treatment of Linear Stochastic Difference Equations," Econometerica, Vol. II, pp. 173-220, 1943.

[2] L. Ljung, "On the Consistency of Prediction Error Identification Methods," in System Identification: Advances and Case Studies,

R.M. Mehra and D.G. Lainotis, Eds., Academic Press, 1976.

[3] P.E. Cains, "Stationary Linear and Nonlinear System Identification and Predictor Set Completeness," IEEE Trans. Aut. Cont., Vol. AC-23, pp. 583-594, 1978.

[4] B.P. Stigum, "Asymptotic Properties of Dynamic Stochastic Parameter Estimates," T. of Multivariable Anal., Vol. 4, pp. 351-381, 1974.

[5] T. Sternby, "On Consistency for the Method of Least Squares Using Martingale Theory," IEEE Trans. Aut. Cont., Vol. AC-22, No. 3, 1977.

[6] L. Ljung, "On Positive Real Transfer Functions and the Convergence of Some Recursive Schemes," IEEE Trans. Aut. Cont., Vol. AC-22, No. 4, 1977.

[7] I.D. Landau, "Unbiased Recursive Identification Using Model Reference Adaptive Techniques," IEEE Trans. Aut. Cont., Vol AC-21, 1976.

[8] I.D. Landau, "Elimination of the Real Positivity Condition in the Design of Parallel MRAS," IEEE Trans. Aut. Cont., Vol. AC-23, 1978.

[9] C.W. Burrill, Measure Integration and Probability, McGraw Hill, Inc., 1972.

[10] E. Fogel, "The Method of Least Squares is a Minimization Procedure - On the Consistency of Least Square Procedures in the Light of This Fact," submitted for publication.

[11] E. Fogel and D. Graupe, "Convergence of Least Squares Identification Algorithms Applied to Unstable Stochastic Process," Int. J. Sys. Sci., Vol. 8, No. 6, 1977.

[12] D. Graupe and E. Fogel, "Convergence of Least Squares Parameter Estimates of Weakly Stationary Time Series Models Driven by Uncorrelated Processes," Int. J. Sys. Sci., Vol. 8, No. 1, 1977.

[13] E. Fogel and D. Graupe, "Mean Convergence of Least Squares Identification of White Noise Driven Time Series Models," Int. J. Sys. Sci., 1977.

SOME RESULTS ON THE IDENTIFIABILITY PROBLEM WITH APERIODIC SAMPLING

M. DE LA SEN* AND S. DORMIDO**
DPTO. DE AUTOMATICA E INFORMATICA. FACULTAD DE CIENCIAS FISICAS
* UNIVERSIDAD DE BILBAO, BILBAO (SPAIN)
** UNIVERSIDAD COMPLUTENSE. MADRID (SPAIN)

Abstract.- The local parametric identifiability problem, in a linear and invariant dynamical system, for small parameter variations around its nominal values, may be reduced to an algebraic linear equation system, in a similar way to the resolution of the state observability problem, by measuring the difference between the output vectors of the parameter perturbed system and the nominal one in an appropriate set of sampling instants.

I. INTRODUCTION

One of the fundamental questions of system parameter identification is whether or not the parameters in a particular mathematical model can be uniquely estimated from a given set of experimental observations. This question is the essence of what is termed "identifiability" [1] .

In cases where there is physical knowledge of system matrices and unknown parameter values, it is often preferable to use physically justified parametrization. Grewall and Glover [4] treat this problem in terms of a distinguishability of a pair of parameter values in a parameter set, and algebraic conditions for identifiability of linear time-invariant and nonlinear systems are derived.

Reid [5] , [7] , takes a sensitivity approach to identifiability in linear time-invariant systems. This approach is motivated by the fundamental association between the system output parameter sensitivities and system parameter identification capability. The method is inherently deterministic and local in nature, but for the common case of systems corrupted by Gaussian white measurement noise, the condition for "sensitivity" identifiability is fully equivalent to the condition that the local information matrix, in a stochastic sense, be positive definite.

Troch [2] states a general formulation for the controllability and observability properties in linear and time-invariant dynamical systems, when a set of appropriate arbitrary sampling instants is chosen (under not very restrictive conditions for this set). In this way, controllability and observability of the continuous system are held and the problem reduces to resolve an algebraic linear equation system. The general formulation above, is used to obtain an adecuate transmission of absolute measuring errors on the output vector towards the result, if adecuate sampling instants are used.

S. Dormido and M. de la Sen [11] study the optimal transmission of relative measuring errors in the observability problem using Numerical Analysis concepts.

In this paper, we state the connections between the general formulation above and the "sensitivity" identifiability concept [5], [7], which permits to reduce the local parametric identifiability problem to an algebraic linear equation system, as it was an observability problem, and by an appropriate choice of the sampling instants, the transmission of possible absolute or relative measuring errors on the output vector, can be minimized. In this way, cost functions or Numerical Analysis concepts may be used.

II. PROBLEM FORMULATION

In a very general setting [5], consider a dynamical system which has a mathematical model output $\underline{y}(\underline{p})$ from the Hilbert space Y. Assume the model output to be parametrized by an unknown, constant, parameter vector $\underline{p} \in R^r$, and $\underline{y}(\underline{p})$ to be everywhere continuously differentiable with respect to each parameter component p_i, i=1, 2,..., r. In a first-order approximation, if v^i denotes the individual parameter sensitivities evaluated at the nominal $\underline{p}_o \in R^r$, $\Delta \underline{p}$ (local variations in the nominal parameter vector when terms of second and higher order in $\Delta \underline{p}$ have been deleted) may be uniquely estimate from $\Delta \underline{y}(.)$ (local variations in the output vector) as it is shown in Definition 1.

Definition 1 [5] : A system will be termed "sensitivity identifiable" at $\underline{p}_o \in R^r$, if $V \in L_c(R^r, \underline{Y})$ is one-to-one, where $V = [v^1, v^2, \ldots, v^r] \in L_c(R^r, \underline{Y})$. The sensitivity nonidentificable subspace is the null space of V, while the sensitivity identifiable subspace is its orthogonal complement.

Definition 2 : A system will be termed "sensitivity identifiable by means of aperiodic sampling" at $\underline{p}_o \in R^r$, if, being "sensitivity identifiable" in the sense given in Definition 1, there is at least an appropriate finite set of sampling instants such that the mapping of the local variations at $\underline{p}_o$ into the local variations in the set of output samples is one-to-one.

Now, let the linear and time-invariant dynamical system be

$$\begin{aligned} &\dot{\underline{x}}(t) = A\underline{x}(t) + \underline{b}\,u(t) \\ &\underline{y}(t) = C\underline{x}(t) \\ &\underline{x}(0) = \underline{x}_o,\ t \in [0, t_f] \\ &\underline{x}(t) \in R^n,\ \underline{y}(t) \in R^s,\ \underline{u}(t) \in R \end{aligned} \qquad (1)$$

where each of the system matrices A, $\underline{b}$ and C and the initial condition vector $\underline{x}_o$ are parametrized by a r-dimensional real parameter vector $\underline{p} \in R^r$. The only conditions assumed on A, $\underline{b}$, C and $\underline{x}_o$ are that they are real, bounded and continuously differentiable with respect to each componet p_i of the parameter vector $\underline{p}$ at $\underline{p}_o$, for i=1, 2,....., r.

Definition 3: The zero-input identifiability-index of system (1) (u=0 for all t) is the positive integer number

$$r_{z.i.} = \min_{1 \leq k \leq 2n} \left\{ k \;/\; \operatorname{rank} \begin{bmatrix} (C\,\underline{x}_o)_{(1)} & \cdots\cdots\cdots & (C\,\underline{x}_o)_{(r)} \\ (C\,A\underline{x}_o)_{(1)} & \cdots\cdots\cdots & (C\,A\underline{x}_o)_{(r)} \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ (C\,A^{k-1}\,\underline{x}_o)_{(1)} & \cdots\cdots & (C\,A^{k-1}\,\underline{x}_o)_{(r)} \end{bmatrix} = r \right\}$$

Definition 4: The initial zero-state identifiability-index of system (1) ($\underline{x}_o = \underline{0}$) is the positive integer number

$$r_{z.s.} = \min_{1 \leq k \leq 2n} \left\{ k \;/\; \operatorname{rank} \begin{bmatrix} (C\,\underline{b})_{(1)} & \cdots\cdots\cdots & (C\,\underline{b})_{(r)} \\ (C\,A\,\underline{b})_{(1)} & \cdots\cdots\cdots & (C\,A\,\underline{b})_{(r)} \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ (C\,A^{k-1}\,\underline{b})_{(1)} & \cdots\cdots & (CA^{k-1}\underline{b})_{(r)} \end{bmatrix} = r \right\}$$

In both definitions, 3 and 4, the subscript (i) denotes partial differentiation with respect to p_i evaluated at $\underline{p}_o$ (the nominal parameter vector).

Theorem 1: Let the system (1) be free (u=0, for all t) and locally "sensitivity identifiable" as indicated in Definition 1. Then it is locally "sensitivity identifiable by means of aperiodic sampling", as indicated in Definition 2, if there is at least a 2n-tupel of positive real numbers (the sampling instants) $0 \leq t_1 < t_2 \;\ldots\; < t_{2n}$ (with $t_{2n} \leq t_f$) fulfilling the condition $\det\,(a_i\,(t_k)) \neq 0$ for $i = 0, 1, \ldots, 2n-1$ and $k = 1, 2, \ldots, 2n$; where the $a_i(.)$ are linear combinations of terms such $t^j\, e^{\lambda_k t}$, λ_k is an arbitrary eigenvalue of the matrix A, for $k = 1, 2, \ldots, \sigma$, (σ is the number of different eigenvalues of A) and j varies from zero to twice the multiplicity of λ_k in the charasteristic polynomial less one.

Proof: we define the matrices

$$J_j^* = \begin{bmatrix} (C_1\,A^{j-1}\,\underline{x}_o)_{(1)} & \cdots\cdots & (C_1\,A^{j-1}\,\underline{x}_o)_{(r)} \\ (C_2\,A^{j-1}\,\underline{x}_o)_{(1)} & \cdots\cdots & (C_2\,A^{j-1}\,\underline{x}_o)_{(r)} \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ (C_s\,A^{j-1}\,\underline{x}_o)_{(1)} & \cdots\cdots & (C_s\,A^{j-1}\,\underline{x}_o)_{(r)} \end{bmatrix}_{S\times r} \qquad j=1,2,\ldots,2n \qquad (2a)$$

$$J^{*} = \begin{bmatrix} J_1^{*} \\ J_2^{*} \\ \vdots \\ J_{2n}^{*} \end{bmatrix}_{2ns \times r} \qquad F = \begin{bmatrix} \dfrac{\partial \underline{y}(t_1)}{\partial \underline{p}} \\ \dfrac{\partial \underline{y}(t_2)}{\partial \underline{p}} \\ \vdots \\ \dfrac{\partial \underline{y}(t_{2n})}{\partial \underline{p}} \end{bmatrix}_{2ns \times r} \qquad (2b)$$

and

$$\Lambda = \begin{bmatrix} \mathrm{Diag}(a_0(t_1),\overset{(s}{\ldots}, a_0(t_1)) & \ldots & \mathrm{Diag}(a_{2n-1}(t_1),\overset{(s}{\ldots}, a_{2n-1}(t_1)) \\ \mathrm{Diag}(a_0(t_2),\overset{(s}{\ldots}, a_0(t_2)) & \ldots & \mathrm{Diag}(a_{2n-1}(t_1),\overset{(s}{\ldots}, a_{2n-1}(t_1)) \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ \mathrm{Diag}(a_0(t_{2n}),\overset{(s}{\ldots}, a_0(t_{2n})) & \ldots & \mathrm{Diag}(a_{2n-1}(t_{2n}),\overset{(s}{\ldots}, a_{2n-1}(t_{2n})) \end{bmatrix}_{2ns \times 2ns}$$

where the $a_i(.)$ have been defined above, (see, for example [5]), the row vectors in the matrix F are obtained by partial differentiation of the output vector with respect to the parameter vector, evaluated at $\underline{p}_o$, and C_j denotes the jth row vector of the matrix C.

It is clear that, in a first-order approximation for the output variations

$$(\Delta \underline{y}(t_1),\ \Delta \underline{y}(t_2), \ldots,\ \Delta \underline{y}(t_{2n}))^t = F\, \Delta \underline{p} \qquad (3)$$

where the $\Delta \underline{y}(.)$ and $\Delta \underline{p}$ denote the local variations in the output vector and the parameter vector, respectively; and $(.)^t$ denotes the transpose of the vector or matrix (.).

The coefficient matrix can be written from (2) as

$$F = \Lambda\, J^{*} \qquad (4)$$

Obviously, if rank (F) = r, $\Delta \underline{p}$ can be uniquely estimated from the set $\Delta \underline{y}(t_1)$, $\Delta \underline{y}(t_2)$, ..., $\Delta \underline{y}(t_{2n})$ following (3) . Therefore, since the system is "sensitivity identifiable" (i.e. rank $(J^{*})=r$), it is "sensitivity identiable by means of aperiodic sampling" too if the condition $\det(a_i(t_k)) \neq 0$, for $i = 0, 1, \ldots, 2n-1$ and $K = 1, 2, \ldots, 2n$; is fulfilled.

Theorem 2: Let the system (1) be, under the same assumptions of theorem above and let $n_1, n_2, \ldots, n_s$ be a set of positive integers fulfilling $\sum_{i=1}^{s} n_i = r$ and $2ns \gtrsim r \gtrsim s$

$$\frac{\partial}{\partial \underline{p}}(C_1\underline{x}_o),\ldots,\ \frac{\partial}{\partial \underline{p}}(C_1A^{n_1-1}\underline{x}_o),\ldots,\ \frac{\partial}{\partial \underline{p}}(C_s\underline{x}_o),\ldots,\frac{\partial}{\partial \underline{p}}(C_sA^{n_s-1}x_o) \tag{5}$$

are r linearly independent row vectors evaluated at p_o, then, the following holds

i) $n_1, n_2,\ldots, n_s$ can be chosen in such a way that

$$\text{Max}\ \{ n_k,\ k= 1, 2,\ldots, s \} = r_{z.i.} \tag{6}$$

ii) In a first-order approximation, the local variations in the parameter vector may be uniquely estimated from the set $\triangle y_1(t_1),\ldots, \triangle y_1(t_{n_1}),\ldots, \triangle y_s(t_1),\ldots, \triangle y_s(t_{ns})$, provided $t_1,t_2,\ldots, t_{n_s}$ fulfill the condition $\det(\text{Diag}\,(\varphi_1, \varphi_2,\ldots, \varphi_s)\,\Psi) \neq 0$, where $\triangle y_i(.)$ denotes the local variation at the ith component of the output vector; and the φ_i for $i= 1,2,\ldots, s$, and Ψ will be defined in (8) and (10).

Proof:

$$E_j^{*'} = \begin{bmatrix} (C_j\,\underline{x}_o)_{(1)} & \cdots\cdots & (C_j\,\underline{x}_o)_{(r)} \\ (C_j\,A\,\underline{x}_o)_{(1)} & \cdots\cdots & (C_j\,A\,\underline{x}_o)_{(r)} \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ (C_jA^{n_j-1}\,\underline{x}_o)_{(1)} & \cdots\cdots & (C_j\,A^{n_j-1}\,\underline{x}_o)_{(r)} \end{bmatrix}_{n_j\times r} \qquad j=1,2,\ldots,s$$

and

$$E^{*'} = \begin{bmatrix} E_1^{*'} \\ E_2^{*'} \\ \vdots \\ E_s^{*'} \end{bmatrix}_{r\times r} \qquad E^{*} = \begin{bmatrix} (C_1\,\underline{x}_o)_{(1)} & \cdots\cdots & (C_1\,\underline{x}_o)_{(r)} \\ \cdots & \cdots & \cdots \\ (C_1\,A^{2n-1}\underline{x}_o)_{(1)} & \cdots & (C_1\,A^{2n-1}\underline{x}_o)_{(r)} \\ \cdots & \cdots & \cdots \\ (C_s\,A^{2n-1}\underline{x}_o)_{(1)} & \cdots & (C_s\,A^{2n-1}\underline{x}_o)_{(r)} \end{bmatrix}_{2ns\times r} \tag{7}$$

According to (5), $E^{*'}$ is non-singular and there is a matrix Ψ, uniquely definited, which verifies

$$E^{*} = \Psi\, E^{*'} \tag{8}$$

it is clear that, in a first-order approximation, for the nominal parameter vector variations

$$\left[\Delta y_1(t_1),\ldots,\ \Delta y_1(t_{n1}),\ldots,\ \Delta y_s(t_1),\ldots,\ \Delta y_s(t_{ns})\right]^t =$$

$$= \text{Diag}\,(\varphi_1,\ \varphi_2,\ldots,\ \varphi_s)\,\Psi\, E^{*'}\,\Delta p \tag{9}$$

where $\triangle \underline{p}$ is the perturbation parameter vector and

$$\varphi_i = \begin{bmatrix} a_0(t_1),\ a_1(t_1),\ldots,\ a_{2n-1}(t_1) \\ a_0(t_2),\ a_1(t_2),\ldots,\ a_{2n-1}(t_2) \\ \cdots\cdots\cdots\cdots\cdots \\ \cdots\cdots\cdots\cdots\cdots \\ a_0(t_{ni}),\ a_1(t_{ni}),\ldots,\ a_{2n-1}(t_{ni}) \end{bmatrix}_{n_i \times 2n} \qquad i=1,\ 2,\ldots,\ s \tag{10}$$

It is obvious from Definition 3 that the $r_{z.i.}$ index may be taken in the way indicated in (6). From (9), we obtain that the system is locally identifiable if the sampling instants fulfill the condition

$$\det\left(\mathrm{Diag}\left(\varphi_1, \varphi_2, \ldots, \varphi_s \right) \psi \right) \neq 0 \tag{11}$$

Theorem 3: Let the system (1) be forced with initial zero-state ($\underline{x}_0=\underline{0}$) and locally "sensitivity identifiable". Then, it is locally "sensitivity identifiable by means of aperiodic sampling "too, if there is at least a 2n-tupel of positive real numbers (the sampling instants) $0 < t_1 < t_2 \ \ldots\ldots < t_{2n}$ fulfilling the condition $\det(b_i(t_k)) \neq 0$ for $i = 0, 1,\ldots,2n-1$ and $k = 1,2,\ldots, 2n$, where the $b_j(t_i)$ are linear combinations of terms such $\int_0^{t_i} a_j\,(t_i-s)\ u(s)\ ds$, where the $a_j(.)$ are defined as in the theorem above.

Proof: we define the matrices

$$J_j^{*} = \begin{bmatrix} (C_1\, A^{j-1}\, \underline{b})_{(1)} & \ldots\ldots & (C_1\, A^{j-1}\, \underline{b})_{(r)} \\ (C_2\, A^{j-1}\, \underline{b})_{(1)} & \ldots\ldots & (C_2\, A^{j-1}\, \underline{b})_{(r)} \\ \cdots\cdots & \cdots\cdots & \cdots\cdots \\ \cdots\cdots & \cdots\cdots & \cdots\cdots \\ (C_s\, A^{j-1}\, \underline{b})_{(1)} & \ldots\ldots & (C_s\, A^{j-1}\, \underline{b})_{(r)} \end{bmatrix}_{s \times r} \qquad j=1,2,..2n \tag{12}$$

$$J^{*} = \begin{bmatrix} J_1^{*} \\ J_2^{*} \\ \vdots \\ J_{2n}^{*} \end{bmatrix}_{2ns \times r} \qquad F = \begin{bmatrix} \dfrac{\partial \underline{y}(t_1)}{\partial \underline{p}} \\ \dfrac{\partial \underline{y}(t_2)}{\partial \underline{p}} \\ \vdots \\ \dfrac{\partial \underline{y}(t_{2n})}{\partial \underline{p}} \end{bmatrix}_{2ns \times r}$$

and

$$\Lambda = \begin{bmatrix} \mathrm{Diag}(b_0(t_1), \overset{(s}{\ldots}, b_0(t_1)) & \ldots & \mathrm{Diag}(b_{2n-1}(t_1), \overset{(s}{\ldots}, b_{2n-1}(t_1)) \\ \mathrm{Diag}(b_0(t_2), \overset{(s}{\ldots}, b_0(t_2)) & \ldots & \mathrm{Diag}(b_{2n-1}(t_2), \overset{(s}{\ldots}, b_{2n-1}(t_2)) \\ \cdots & \cdots & \cdots \\ \mathrm{Diag}(b_0(t_{2n}), \overset{(s}{\ldots}, b_0(t_{2n})) & \ldots & \mathrm{Diag}(b_{2n-1}(t_{2n}), \overset{(s}{\ldots}, b_{2n-1}(t_{2n})) \end{bmatrix}_{2ns\times 2ns}$$

It is clear for the output variations, in a first-order aporoximation, that

$$[\Delta \underline{y}(t_1),\ \Delta \underline{y}(t_2), \ldots,\ \Delta \underline{y}(t_{2n})\ ]^t = F\, \Delta\, \underline{p} \tag{13}$$

where the $\Delta\, y(.)$ and $\Delta\, \underline{p}$ denote the local variations in the output vector and the parameter vector, respectively.

By similar considerations to theorem 1, we can conclude that if the system (1) is "sensitivity identifiable", it is "sensitivity identifiable by means of aperiodic samoling" too, if the condition $\det\,(b_i(t_k)) \neq 0$, for $i=0,1,\ldots,\ 2n-1$ and $k=1,2,\ldots,\ 2n$; is fulfilled.

Since the functions $a_i(.)$ all tend to zero with increasing time (assuming an asintotically stable system), for continued identifiability in the steady state, one must have "sufficient excitation" of the control input to maintain linear idependence of the functions $b_i(.)$ and therefore, this is also a necessary condition for "identifiability by means of aperiodic sampling" for al $t_j \in \{t_1, t_2, \ldots, t_{2n}\}$.

Theorem 4: It is applied to forced systems and it is similar to theorem 2.

Remark 1: Note that under the conditions given in theorems 2 and 4 the $r_{z.i.}$ and $r_{z.s.}$ identifiability indices represent le minimum number of sampling instants necesaary to identify the unforced and initial zero-state forced systems respectively. Otherwise it may be possible to identify the system with a fewer number of sampling instants than the corresponding identifiability indices

Remark 2: The sampling instants can be chosen arbitrary, but fulfilling the not very strong conditions given in the theorems above. In order to minimize the transmission of measuring errors on the output vector, appropriate cost functions or the matrix condition number concept of Numerical Analysis can be used.

Remark 3: By application of this formulation, we assure that the mapping of the first-order local variations in the parameter vector into an appropriate set of local differences in the output vector in one-to-one, for the unforced and zero-state forced systems respectively, and the problem reduces to solve an algebraic system of linear equations.

Obviously the number of parameters must not be greater than 2ns.

III. EXAMPLE

Now, let the system (1) be free, and assume that the matrix A has been transformed to the real canonical form and that the locally disturbed components of the parameter vector are the eigenvalues of A, which are all real and distinct. Assume also that rank(C) = n and $r_{z.i.} = 2$.

In a first-order approximation

$$\Delta \underline{y}(t) \doteq C \text{ Diag } \left(t\, e^{\lambda_1 t} x_{o1},\ t\, e^{\lambda_2 t} x_{o2}, .., \ t\, e^{\lambda_n t} x_{on}\right) \Delta \underline{p} =$$

$$= G \, \Delta \underline{p} \tag{14}$$

where $\underline{x}_o = (x_{o1}, x_{o2}, \ldots, x_{on})^t$ with $x_{oi} \neq 0$ for all i

From (14), we obtain

$$\Delta \underline{p} \doteq G^{-1} \, \Delta \underline{y}(t)$$

Now, we choose as cost function to obtain an optimal transmission of measuring errors on the output vector

$$J = \left\| \frac{\Delta \underline{p}}{\Delta \underline{y}(t)} \right\|_E^2 = \sum_{k=1}^{n} (G^{-1} \, \underline{e}_k)^2 = \sum_{k=1}^{n} \langle G^{-1} \underline{e}_k , \, G^{-1} \underline{e}_k \rangle =$$

$$= \text{trace} \left((G^{-1})^t \, G^{-1} \right) = \frac{1}{t^2} \sum_{i,j=1}^{n} (C_{ij}^{-1})^2 \, \frac{e^{-2\lambda_i t}}{x_{oi}^2} \tag{15}$$

where $\| \cdot \|_E$ denotes the euclidean matrix norm, $\langle . \, , \, \rangle$, denotes the inner product, C_{ij}^{-1} is an arbitrary element of the inverse of the matrix C, and k is the k-unity vector.

The optimal measuring time, which minimizes the cost function J, may be found by solving

$$\frac{1}{t^3} \sum_{i,j=1}^{n} (C_{ij}^{-1})^2 \, (1 + \lambda_i t) \, \frac{e^{-2\lambda_i t}}{x_{oi}^2} = 0 \tag{16}$$

It is clear from (15) and (16) that

- If all the eigenvalues of A are positive, (16) has no acceptable solution, and the measuring time must ve taken as large as possible, in order to minimize (15). A relative minimum cannot be found.

- If not all the eigenvalues of A have the same sign, a measuring time which fulfills (16) may be found. This solution may or not to have a real physical meaning.

- If all the eigenvalues of A are negative, there is always a measuring time which verifies (16).

We conclude that only if the system is stable, there is always an optimal measuring time which minimizes (15).

REFERENCES

[1] P. EYKLOFF "System Identification", John Wiley and Sons, London 1974.

[2] I. TROCH "Sampling with arbitrary choice of the sampling instants". Automatica, vol 9, pp 117-124, 1973.

[3] T.C. HSIA."On least squares algorithms for system parameter identification". IEEE Trans. Automatic Control, vol AC-21, nº 1, pp 104-108, February 1976.

[4] M.S. GREWALL and K. GLOVER "Identifiability of linear and nonlinear dynamical system". IEEE Trans. Automatic Control, vol AC 21, nº 6, pp 833-837, December 1976.

[5] J.G. REID "Structural Identiability in linear time-invariant systems". IEEE Trans. Automatic Control, Vol AC-22, nº 2, pp 242-246, April 1977.

[6] A. Thowsen "Identifiability of dynamic systems". Int. Journal of Systems Science, col 9, nº 7, pp 813-825, July, 1978.

[7] J.G. REID "Sensitivity Operators and associated system concepts for linear dynamic systems" Ph. D. Diss. Air Force Institute of Technology, Ohio, 1976.

[8] F. M. FISHER "The identification problem in econometrics". Mc. Graw-Hill, New York, 1966.

[9] F.M. FISHER "Generalization of the rank and order conditions for identifiability. "Econometrica, vol 27, nº 3, 1959.

[10] T.J. ROTHENBERG "Identification in parametric models". Econometrica, vol 39, nº 3, May 1971.

[11] S.DORMIDO and M. DE LA SEN "A note on the transmission of relative errors in the observability problem". IEEE Trans. Automatic Control, to be published.

[12] M.DE LA SEN "Propiedades Fundamentales de los Sistemas Dinámicos con elección aperiódica de los instantes de muestreo". Tesis Doctoral, Universidad de Bilbao, Junio 1979.

DISCRETE SPECTRAL ESTIMATION WITH OBSERVERS*

G. H. HOSTETTER
Electrical Engineering Department
California State University, Long Beach, Ca. 90840

ABSTRACT

This paper presents a new approach to Fourier spectral analysis in which ongoing spectral estimates are generated in real time, with arbitrary convergence rates.

Both the continuous-time and the discrete-time filters are developed. Results are presented within the unifying framework of observer theory.

Fifth order system examples are given which demonstrate representative performance and indicate the ability of these spectral observers to respond to non-bandlimited signals or their samples and to track time-varying harmonic composition.

I. INTRODUCTION

There exist in practice many circumstances where it is desired to estimate the spectral composition of a signal. This paper presents new spectral analysis methods which promise advantages over conventional swept-filter analog and digital Fourier transform techniques. Recursive spectral estimates may be generated in real time, with arbitrary rates of convergence. Both continuous-time and discrete-time filters for spectral estimation are developed, and numerical examples of each are given.

The observer theory viewpoint adopted here, while not absolutely essential, is extremely convenient for relating these results to those of others and for easily bringing to bear powerful general knowledge regarding system structure and eigenvalue placement [1 - 4].

These results are a natural extension of the work of many researchers who have considered observing unknown system signals by modelling them with differential equations. This approach dates from the work of Johnson [5-6] and Davison [7-8], although without explicit connection to observers at that time. Bryson and Luenberger [9], Belanger [10], and Young and Willems [11] considered similar problems. Hostetter and Meditch related Davison's work to observers [12] and investigated their algebraic structure and properties in detail [13-14].

II. CONTINUOUS-TIME FOURIER ANALYSIS

Outline of Continuous-Time Observer Theory

For the linear, time-invariant $(n)^{th}$ order system

$$\dot{x} = Ax + Bu$$

$$y = Cx, \qquad (1)$$

another system

---

* Research supported in part by a grant from the California State University, Long Beach Foundation.

$$\dot{z} = Fz + Gy + Hu$$

is said to be an identity observer [1-3] or, equivalently, a state estimator [4] of (1), providing there exists a $z(0)$, depending upon $x(0)$, such that $z(t) = x(t)$.

For any initial conditions, the difference between $x(t)$ and $z(t)$ is given by

$$\dot{x} - \dot{z} = (A-GC)x - Fz + (B-H)u.$$

with the choices

$$F = A - GC$$

$$H = B,$$

then

$$\dot{x} - \dot{z} = F(x-z),$$

$$x(t) - z(t) = [\exp(Ft)][x(0)-z(0)].$$

For fixed A and C, the eigenvalues of F can be chosen arbitrarily by an appropriate choice of G, providing that the pair (A, C) is completely observable [15-21, 1-4]. An observer estimate, $z(t)$, may then be made to converge arbitarily quickly to $x(t)$, within the practical limitations of noise and imperfect knowledge of the system parameters.

Luenberger showed that if a completely observable system has m linearly independent outputs, an identity observer with arbitrary eigenvalues and reduced order n-m, of the form

$$\dot{z} = Fz + Gy + Bu$$

$$w = Lz + My,$$

where w is the observer output, may be constructed [22-26, 1-4].

System Representation for an Arbitrary Bandlimited Periodic Function

An arbitrary bandlimited periodic function $y(t)$ may be expressed in the form

$$y(t) = d_o + \sum_{n=1}^{N} (\alpha_n \cos \frac{n2\pi t}{T} + \beta_n \sin \frac{n2\pi T}{T}) \qquad (2)$$

where T is the period of the function, N is the harmonic number of the highest frequency present in $y(t)$, and d_o, the α's and the β's are the function's Fourier coefficients [27-28].

A linear, time-invariant, homogeneous differential equation with characteristic equation

$$s[s^2 + (\frac{2\pi}{T})^2][s^2 + (\frac{4\pi}{T})^2] \cdots [s^2 + (\frac{N2\pi}{T})^2] = 0$$

has general solution of the form of (2), where d_o, the α's and the β's are arbitrary constants. The general solution of such an equation is an arbitrary periodic function with period T and bandlimit beyond the N^{th} harmonic.

A convenient state variable form for this homogenous system is the following, in which the constant and each harmonic term in the response are decoupled from one another in the state equations:

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \vdots \\ \dot{x}_{2N-1} \\ \dot{x}_{2N} \\ \dot{x}_{2N+1} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ -(\frac{2\pi}{T})^2 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 & 0 \\ 0 & 0 & -(\frac{4\pi}{T})^2 & 0 & \cdots & 0 & 0 \\ \vdots & & & & & & \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & 0 & 0 & \cdots & -(\frac{N2\pi}{T})^2 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_{2N-1} \\ x_{2N} \\ x_{2N+1} \end{bmatrix}$$

$$= Ax,$$

$$y = [1 \quad 0 \quad 1 \quad 0 \quad \cdots \quad 1 \quad 0 \quad 1]\, x$$

$$= c^T x \qquad (3)$$

Observing the Harmonics

Although a bandlimited periodic signal y(t) may not actually be produced by the system (3), considering it to be the output of such a system and driving an identity observer of that system with y(t) will cause the observer state to converge to the state of (3).

Were the system (3) producing y(t), the individual harmonic components of y are the signals

$$x_1(t) = \alpha_1 \cos \frac{2\pi t}{T} + \beta_1 \sin \frac{2\pi t}{T}$$

$$x_2(t) = \frac{dx_1}{dt} = \frac{2\pi}{T} (\beta_1 \cos \frac{2\pi t}{T} - \alpha_1 \sin \frac{2\pi t}{T})$$

$$x_3(t) = \alpha_2 \cos \frac{4\pi t}{T} + \beta_2 \sin \frac{4\pi t}{T}$$

$$\vdots$$

$$x_{2N-1}(t) = \alpha_N \cos \frac{N2\pi t}{T} + \beta_N \sin \frac{N2\pi t}{T}$$

$$x_N(t) = \frac{dx_{2N-1}}{dt} = \frac{N2\pi}{T} (\beta_N \cos \frac{N2\pi t}{T} - \alpha_N \sin \frac{N2\pi t}{T})$$

$$x_{2N+1}(t) = d_o.$$

Hence, an observer may be used to precisely separate the harmonics of a bandlimited periodic signal.

If the individual quadrature harmonic amplitudes are then detected, the observer performs a Fourier analysis of the waveform $y(t)$. The rate of convergence of the observer estimates to the true harmonic components may be arbitrarily rapid.

Alternatively, an observer may be used to estimate the initial conditions [29-30] of the system (3), which are easily related to d_o, the α's and the β's.

If samples of $y(t)$ are available, they may be used to drive a continuous-time observer [31-32] of (3).

At steady state, input frequencies above the highest modelled (foldover) frequency will cause components at these frequencies to be superimposed upon the spectral estimates. It is then possible to lowpass filter to remove aliasing distortion after the spectral signals are generated by the observer.

When lowpass prefiltering is performed, there is possibility of observing some of the filter dynamics as well as the signal to be analyzed. Imperfect filtering such as passband ripple may thus be compensated by the observer. The observer (if its order is not reduced) provides additional lowpass filtering of unwanted high frequencies, and there is the possibility of placing the observer eigenvalues for optimum bandlimiting rather than for best transient response.

Continuous-Time Example

Consider a bandlimited periodic function of the form

$$y(t) = d_o + \alpha_1 \cos t + \beta_1 \sin t + \alpha_2 \cos 2t + \beta_2 \sin 2t \tag{4}$$

Any signal of this form can be produced by the homogeneous system (3) with $T = 2\pi$.

The system

$$\dot{z} = Fz + gy(t) = \begin{bmatrix} -13.33 & 1 & -13.33 & 0 & -13.33 \\ -14.67 & 0 & 13.67 & 0 & -13.67 \\ 10.83 & 0 & 10.833 & 1 & 10.83 \\ -21.33 & 0 & -25.33 & 0 & -21.33 \\ -7.5 & 0 & -7.5 & 0 & -7.5 \end{bmatrix} z + \begin{bmatrix} 13.33 \\ 13.67 \\ -10.83 \\ 21.33 \\ 7.5 \end{bmatrix} y(t)$$

$$w = [1 \quad 0 \quad 1 \quad 0 \quad 1]\, z$$

has five $s = -2$ eigenvalues and has state z which observes the state of (4). This observer was designed using transformations to and from the dual phase variable canonical form [1 -4]. If the signal $y(t)$ drives the observer, $z_1(t)$, $z_3(t)$ and $z_5(t)$ will rapidly converge to the individual harmonic components of $y(t)$.

Experimental Results

Experimental response of the example observer, driven by a signal of the form (4) are presented in figure 1. Both analog and digital simulations were performed for a number of such signals, for which the curves shown are representative. For the analog generation, the periodic signal was generated by synchronized oscillators. The observer consisted of operational amplifier elements connected on an EAI TR-20 analog computer. Digital simulation was performed on an HP 9830 system.

Figure 2 shows the system response to waves with slowly varying harmonic composition. Tracking of the harmonic amplitudes can be very good and applications such as speech and other audio pitch detection come to mind.

III. DISCRETE-TIME FOURIER ANALYSIS

Outline of Discrete-Time Observer Theory

For a constant-coefficient n^{th} order discrete-time system

$$\xi(k+1) = A\xi(k) + Bu(k)$$
$$w(k) = C\xi(k) \tag{5}$$

another related system,

$$\varepsilon(k+1) = \Phi\varepsilon(k) + Kw(k) + Lu(k)$$

is termed an identity observer [33-34] of (5) provided that for every $\xi(0)$, if $\varepsilon(0) = \xi(0)$, then $\varepsilon(k) = \xi(k)$, $k = 0, 1, 2, \ldots$

The choices

$$\Phi = A - KC$$

$$L = H$$

give

$$\xi(k+1) - \varepsilon(k+1) = \Phi[\xi(k) - \varepsilon(k)],$$

and the observer state may be made to converge rapidly to the state of (5) by appropriate choice of the eigenvalues of Φ. If all eigenvalues of Φ are chosen to be zero, the observer is a deadbeat system, with state $\varepsilon(k)$ which converges exactly to the observed system state $\xi(k)$ in n steps or less.

Discrete-Time System Representation

A discrete-time model of the continuous-time system (3) is

$$\xi(k+1) = x\,[(k+1)\Delta t] = [\exp(A\Delta t)]\ \xi(k)$$
$$w(k) = y(k\Delta t) = c^T\xi(k) \tag{6}$$

The system (6) generates samples of the arbitrary bandlimited periodic signal y(t); the state signals are samples of the harmonic components of y(t).

For the Nyquist sampling rate

$$\Delta t = \frac{T}{2N+1}$$

the samples at k = 0, 2N+1, 4N+2, ... are the cosine and sine amplitudes.

Harmonic Observation

A discrete-time observer of (6), whether or not the observed system per se actually exists, can, from independent samples of y(t) accurately estimate independent samples of the harmonics.

For a deadbeat observer, convergence of the observer state to the harmonic model state will occur in (2N+1)steps. At the Nyquist sampling rate, this convergence occurs at the last of the first period's samples, allowing recursive harmonic analysis in real time. For each sample, the observer must execute only a single real number vector-matrix multiplication.

With special canonical forms for the observer, it is conceivable that this procedure could rival the computational efficiency of fast Fourier transform (FFT) methods. If imperfect convergence in (2N+1) steps can be tolerated, so that nonrepeated observer eigenvalues may be used, diagonalization of the observer state coupling matrix Φ greatly reduces the required arithmetic operations. If a design set of observer eigenvalues can be perturbed sufficiently, so as to give simple binary fraction elements for Φ, large savings in the necessary arithmetic precision can be made. It is thereby possible to also significantly reduce roundoff error effects. When the discrete-time spectral observer is used to perform digital Fourier transformation, the input samples will be processed as those of a bandlimited signal. If the input signal is not bandlimited, frequency aliasing will produce the usual distortion. There is, in addition, the ability to easily accommodate sampling rates and numbers of samples other than the Nyquist rate and number.

Nyquist Rate Discrete-Time Example

The example discrete-time harmonic model (6) is, for the Nyquist sampling rate $\Delta t = 2\pi/5$,

$$\xi(k+1) = \begin{bmatrix} 0.309 & 0.951 & 0 & 0 & 0 \\ -0.951 & 0.309 & 0 & 0 & 0 \\ 0 & 0 & -0.809 & 0.294 & 0 \\ 0 & 0 & -1.1755 & -0.809 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \xi(k) \qquad (7)$$

The system

$$\varepsilon(k+1) = \begin{bmatrix} 0.185 & 0.95 & -0.1236 & 0 & -0.1236 \\ -0.57 & 0.309 & 0.38 & 0 & 0.38 \\ 0.32 & 0 & -0.489 & 0.294 & 0.32 \\ 0.47 & 0 & -0.6056 & -0.809 & 0.47 \\ -0.2 & 0 & -0.2 & 0 & 0.8 \end{bmatrix} \varepsilon(k) + \begin{bmatrix} 0.124 \\ -0.38 \\ -0.32 \\ -0.47 \\ 0.2 \end{bmatrix} w(k)$$

$$\gamma(k) = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \end{bmatrix} \varepsilon(k) \qquad (8)$$

is a deadbeat observer of (7), all eigenvalues of Φ being at the origin.

General Rate Discrete-Time Example

For a sampling rate above the Nyquist rate, $\Delta t = 1/2$, the example discrete-time harmonic model (6) is as follows:

$$\xi(k+1) = \begin{bmatrix} 0.878 & 0.48 & 0 & 0 & 0 \\ -0.48 & 0{,}878 & 0 & 0 & 0 \\ 0 & 0 & 0.54 & 0.42 & 0 \\ 0 & 0 & -1.683 & 0.54 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \xi(k)$$

The system

$$\varepsilon(k+1) = \begin{bmatrix} -0.236 & 0.48 & -1.114 & 0 & -1.114 \\ -6.63 & 0.878 & -6.15 & 0 & -6.15 \\ 1.72 & 0 & 2.26 & 0.42 & 1.72 \\ -1.29 & 0 & -2.97 & 0.54 & -1.29 \\ -4.44 & 0 & -4.44 & 0 & -3.44 \end{bmatrix} \varepsilon(k) + \begin{bmatrix} 1.114 \\ 6.15 \\ -1.72 \\ 1.29 \\ 4.44 \end{bmatrix} w(k)$$

$$\gamma(k) = [1 \quad 0 \quad 1 \quad 0 \quad 1]\,\varepsilon(k)$$

is a deadbeat observer of (10). After five steps, two seconds of the input waveform, the observer state will converge to the bandlimited input samples.

Experimental Results

The example Nyquist rate discrete-time observer, driven by digitally generated samples of a signal of the form (4) gave the representative results shown in figure 3. The realization was in software on an HP 2100 system.

In figure 4 is shown the higher sampling rate example response. Convergence to the state of the bandlimited signal occurs before completion of a period of the signal.

IV. CONCLUSION

Techniques for Fourier spectral analysis have been presented and illustrated with numerical examples. Filter computations are performed in a real number state variable format.

The continuous-time filter may be designed for an arbitrarily rapid convergence rate. It is useful even with input signals which are not bandlimited and in tracking slowly varying spectral composition.

The discrete-time fitler is easily designed to accommodate various numbers of discrete frequencies and any sampling rates.

References

1. D. G. Luenberger, "Observing the state of a linear system", IEEE Trans. on Military Electronics, Vol. MIL-8, April 1964, pp. 74-80.

2. ______________, "Observers for multivariable systems", IEEE Trans. Automatic Control, Vol. AC-11, April 1966, pp. 190-197.

3. ______________, "An introduction to observers", IEEE Trans. Automatic Control, Vol. AC-16, Dec. 1971, pp. 596-602.

4. B. D. O. Anderson and J. B. Moore, Linear Optimal Control, Prentice-Hall, Englewood Cliffs, N. J., 1971.

5. C. D. Johnson, "Optimal control of the linear regulator with constant disturbances", IEEE Trans. Automatic Control, Vol. AC-13, Aug. 1968, pp. 416-421.

6. ______________, "Further study of the linear regulator with disturbances--the case of vector disturbances satisfying a linear differential equation", IEEE Trans. Automatic Control, Vol. AC-15, April, 1970, pp. 222-228.

7. E. J. Davison, "Comments on 'Optimal control of the linear regulator with constant disturbances'", IEEE Trans. Automatic Control, Vol. AC-15, pg. 264, April 1970.

8. ______________, "The output control of linear time-invariant multivariable systems with unmeasurable arbitrary disturbances", IEEE Trans. Automatic Control, Vol. AC-17, pp. 621-629, October 1972.

9. A. E. Bryson and D. G. Luenberger, "The synthesis of regulator logic using state-variable concepts", Proc. IEEE, Vol. 58, pp. 1803-1811, November 1970.

10. P. R. Belanger, "Observations and control of linear systems with constant disturbances", IEEE Trans. Automatic Control, Vol. AC-17, pp. 416-421, October 1972.

11. P. C. Young and J. C. Willems, "An approach to the linear multivariable servomechanism problem", Intl. J. Control, Vol. 15, pp. 961-979, 1972.

12. G. H. Hostetter and J. S. Meditch, "Observing systems with unmeasurable inputs", IEEE Trans. Automatic Control, Vol. AC-18, pp. 307-308, June 1973.

13. ______________, "On the generalization of observers to systems with unmeasurable and unknown inputs", Automatica, Vol. 9, pp. 721-724, November 1973.

14. J. S. Meditch and G. H. Hostetter, "Observers for systems with unknown and unaccessible inputs", Intl. J. Control, Vol. 19, pp. 473-480, 1974.

15. W. M. Wonham and C. D. Johnson, "Optimal bang-bang control with quadratic performance index", J. Basic Engr. Trans. ASME, Vol. 86, March 1964, pp. 107-115.

16. C. D. Johnson and W. M. Wonham, "A note on the transformation to canonical (phase-variable) form", IEEE Trans. Automatic Control, Vol. AC-9, July 1964, pp. 312-313.

17. D. G. Luenberger, "Invertible solutions to the operator equation TA-BT=C", Proc. Amer. Math Soc., Vol. 16, Dec. 1965, pp. 1226-1229.

18. ______________, "Canonical forms for linear multivariable systems", IEEE Trans. Automatic Control, Vol. AC-12, June 1967, pp. 290-293.

19. W. M. Wonham, "On pole assignment in multi-input controllable linear systems", IEEE Trans. Automatic Control, Vol. AC-12, Dec. 1967, pp. 660-665.

20. C-T Chen, "A note on pole assignment", IEEE Trans. Automatic Control, AC-13, Oct. 1968, pp. 597-598.

21. E. J. Davison, "On pole assignment in multivariable linear systems", IEEE Tran. Automatic Control, Vol. AC-13, Dec. 1968, pp. 747-748.

22. F. Dellon andP. E. Sarachik, "Optimal control of unstable plants with inaccessible states", IEEE Trans. Automatic Control, Vol. AC-13, Oct. 1968, pp. 491-495.

23. J. B. Pearson, "Compensator design for dynamic optimization", Int. Jour. Control, Vol. 9, No. 4, 1969, pp. 473-482.

24. ______________, and C. Y. Ding, "Compensator design for multivariable linear systems", IEEE Trans. Automatic Control, Vol. AC-14, April 1969, pp. 130-134.

25. J. D. Ferguson and Z. V. Rekasius, "Optimal linear control systems with incomplete state measurements", IEEE Trans. Automatic Control, Vol. AC-14, April 1969, pp. 135-149.

26. B. Gopinath, "On the control of linear multiple input-output systems", Bell System Tech. J., Mar. 1971, pp. 1101-1113.

27. A. Popoulis, The Fourier Transform and Its Applications, McGraw-Hill, New York, 1962.

28. ______________, Signal Analysis, McGraw-Hill, New York, 1977.

29. G. H. Hostetter and J. S. Meditch, "Observers for the computation of initial conditions", Proc. Fourth Symp. Nonlinear Est. Theory & Appl., Sept. 1973, pp. 132-134.

30. J. S. Meditch and G. H. Hostetter, "Estimation of initial conditions in linear systems", Proc. Eleventh Allerton Conf. on Cir. Syst. Theory, October 1973, pp. 226-235.

31. G. H. Hostetter, "Sampled data updating of continuous-time observers", IEEE Trans. Automatic Control, Vol. AC-21, pp. 272-273, April 1976.

32. A. Salama and V. Gourishankar, "Continuous-time observers with delayed sampled data", IEEE Trans. Automatic Control, Vol. AC-23, Dec. 1978, pp. 1107-1109.

33. E. Tse, "On the optimal control of linear sy-tems with incomplete information", Ph.D. thesis and Report ESL-R-412, M.I.T., Jan. 1970.

34. E. Tse and M. Athaus, "Optimal Minimum-order observer-estimators for discrete linear time varying systems", IEEE Trans. Automatic Control, Vol. AC-15, Aug. 1970, pp. 416-426.

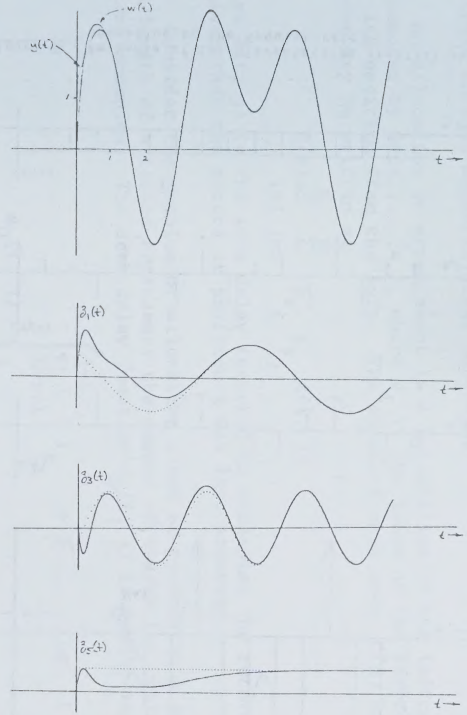


Figure 1: Continuous-time spectral observer response

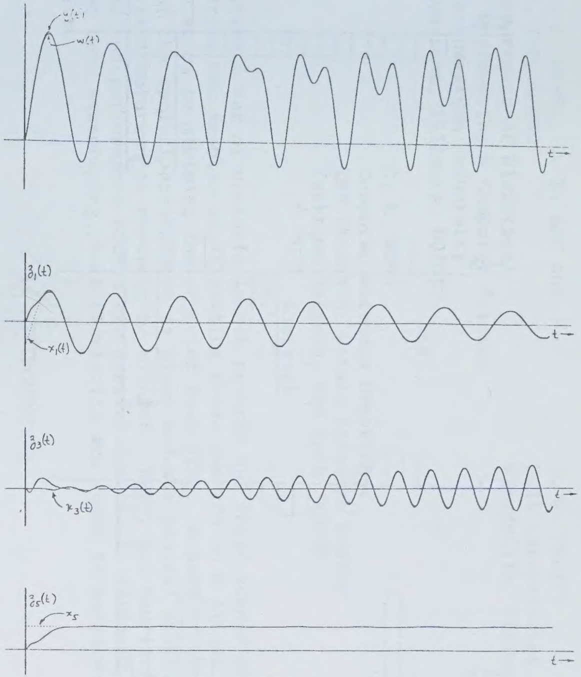


Figure 2: Harmonic tracking by the continuous-time spectral observer

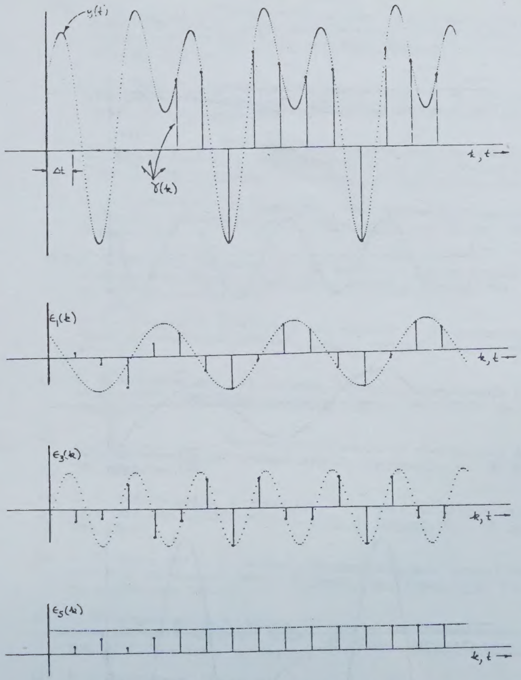


Figure 3: Response of the discrete-time spectral observer operating at the Nyquist rate

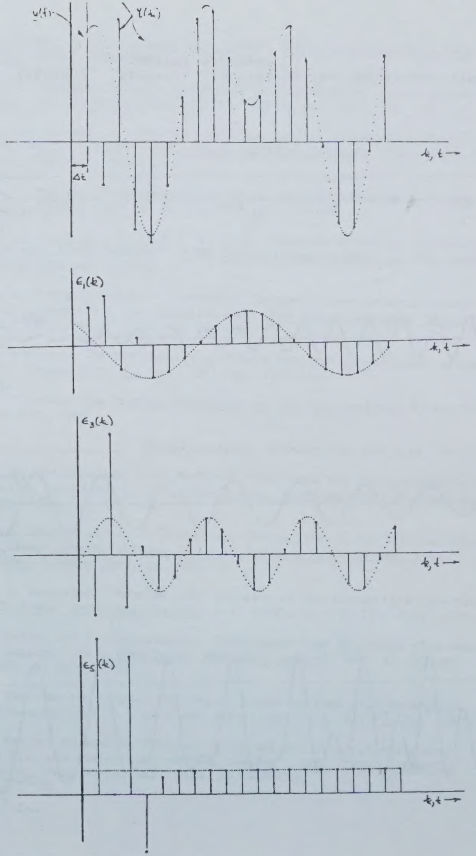


Figure 4: Response of the discrete-time spectral observer operating above the Nyquist rate

RECORD ALLOCATION TO MINIMIZE EXPECTED SEEK DELAY[†]

U. I. GUPTA, D. T. LEE and
J. Y-T. LEUNG
Department of Electrical
Engineering & Computer Science
Northwestern University
Evanston, Illinois 60201

J. W. PRUITT
Bell Laboratories
Naperville, Illinois 60540

and

C. K. WONG
Computer Sciences Department
IBM Thomas J. Watson Research Center
Yorktown Heights, New York 10598

Abstract

Given a set of variable length records and their access probabilities, we address the problem of allocating these records on a linear storage device so as to minimize the expected seek time. A partial characterization of the optimal arrangement is given and the general problem of finding an optimal solution is shown to be NP-hard. Next, two heuristics are considered and performance bounds are derived for them. Although these bounds are not very encouraging, both heuristics are found to perform well in practice.

1. INTRODUCTION

We consider the problem of allocating a set of records on an auxiliary storage device so as to minimize the expected seek delay time. The records are represented by the access probabilities $\{p_1,\ldots,p_n\}$ and the corresponding record lengths $\{l_1,\ldots,l_n\}$. The problem is formulated under the assumption that record requests are served first-in-first-out (FIFO) and that consecutive accesses are independent. The storage devices applicable to our analyses include magnetic disks and tapes.

An allocation or arrangement is a permutation of the indices 1 to n and is denoted by $S=(s_1,\ldots,s_n)$, where s_i is the index of the record assigned to the ith position from the left. The expected seek cost C(S) of a record access may be written as

$$C(S)=\sum_{i=1}^{n}\sum_{j=1}^{n} p_{s_i}\, p_{s_j}\, d(i,\, j), \qquad (1)$$

where d(i, j) is the seek delay incurred in accessing the record at position j given that the record at position i was last accessed.

Consider the problem of allocating a set of records to a magnetic disk. The length of a record represents the number of tracks (or cylinders) the record occupies. The seek delay function $d_D(i, j)$ is easily seen to be

$$d_D(i,\, j) = \begin{cases} \sum\limits_{k=i+1}^{j-1} l_{s_k} + 1 & \text{for} \quad i < j \\ \sum\limits_{k=j}^{i} l_{s_k} - 1 & \text{for} \quad i \geq j \end{cases} \qquad (2)$$

[†] This research was supported in part by the National Science Foundation under Grant MCS-79-04898.

Substituting (2) into (1) and simplifying, the expected seek cost function $C_D(S)$ is given by

$$C_D(S) = \sum_{i=1}^{n} p_{s_i} (l_{s_i} - p_{s_i}) + 2\sum_{i=2}^{n-1} l_{s_i} \Big(\sum_{j=1}^{i-1} p_{s_j}\Big) \Big(\sum_{j=i+1}^{n} p_{s_j}\Big). \qquad (3)$$

If the storage device is a magnetic tape, the length of a record represents units of distance the record occupies. The seek delay function $d_T(i,j)$ is given by

$$d_T(i,j) = \begin{cases} \sum_{k=i+1}^{j-1} l_{s_k} & \text{for } i < j \\ \sum_{k=j}^{i} l_{s_k} & \text{for } i \geq j . \end{cases} \qquad (4)$$

Substituting (4) into (1) and simplifying, we derive the expected seek cost function $C_T(S)$ to be

$$C_T(S) = \sum_{i=1}^{n} p_{s_i} l_{s_i} + 2\sum_{i=2}^{n-1} l_{s_i} \Big(\sum_{j=1}^{i-1} p_{s_j}\Big) \Big(\sum_{j=i+1}^{n} p_{s_j}\Big). \qquad (5)$$

If $C_D(S)$ is expressed in terms of $C_T(S)$, then the formula

$$C_D(S) = C_T(S) - \sum_{i=1}^{n} p_{s_i}^2$$

relates the two problems. Since the difference between the two expected seek cost function is independent of the arrangement, an optimal arrangement, for one problem will be an optimal arrangement for the other. In this paper, we shall restrict our attention to the cost function $C_T(S)$. function $C_T(S)$.

To simplify our notation, we shall assume that the records have been reindexed so that $s_i = i$ for all $1 \leq i \leq n$. With this notation, the expected seek cost function is simply written as

$$C(S) = \sum_{i=1}^{n} p_i l_i + 2\sum_{i=2}^{n-1} l_i \Big(\sum_{j=1}^{i-1} p_j\Big) \Big(\sum_{j=i+1}^{n} p_j\Big). \qquad (6)$$

An optimal arrangement S_o is an arrangement which satisfies $C(S_o) \leq C(S)$ for all possible arrangements S. We are interested in devising fast algorithms which will determine the optimal arrangement for any given set of records.

An efficient algorithm for finding the optimal arrangement of a set of equal-length records has been known for a long time [2,3,5]. The optimal arrangement consists of arranging the records such that the access probabilities form an "organ-pipe arrangement"; i.e. the record with the highest access probability is placed in the middle and the records with successively lower probabilities are placed adjacently on alternate sides. Similarly, the optimal arrangement for a set of equal-access-probability records is one in which the reciprocals of the record lengths form an organ-pipe arrangement. These results seem to indicate that the optimal arrangement for the general case would be one in which the ratios p_i/l_i form an organ-pipe arrangement. However, it turns out that this is not true, and, we shall show in the sequel that finding an optimal arrangement is NP-hard.

2. CHARACTERISTICS OF OPTIMAL ARRANGEMENT

We begin by giving some characterizations of an optimal arrangement. In addition to simplifying our proof of NP-completeness, these results can be quite useful when branch-and-bound techniques are used to obtain exact solutions. First, we derive a formula for the change in cost when two records are interchanged.

Lemma 1 Let S' be obtained from S by interchanging the records at positions t and t', where $1 \leq t < t' \leq n$. Then we have

$$C(S') - C(S) = 2\,[D_1(t,t')R_1(t,t') + D_2(t,t')R_2(t,t')] \tag{7}$$

where $D_1(t,t') = p_t - p_{t'}$,

$$D_2(t,t') = \sum_{i=1}^{t-1} p_i - \sum_{i=t'+1}^{n} p_i ,$$

$$R_1(t,t') = \sum_{i=t+1}^{t'-2} \sum_{j=i+1}^{t'-1} R_3(i,j) ,$$

$$R_2(t,t') = \sum_{j=t}^{t'-1} (R_3(j,t') + R_3(t,j)),$$

and $R_3(u,v) = p_u l_v - p_v l_u$.

Corollary Let S' be obtained from S by interchanging the records at positions t and t + 1, where $1 \leq t \leq n-1$. Then we have

$$C(S') - C(S) = 2\,\left(\sum_{i=1}^{t-1} p_i - \sum_{i=t+2}^{n} p_i\right)(p_t l_{t+1} - p_{t+1} l_t). \tag{8}$$

The record(s) with the highest p_i/l_i ratio play an important role in the optimal arrangement. Our next lemma shows that these records must be placed contiguously in the optimal arrangement. That is, if there are r+1 records with the highest p_i/l_i ratio, then they must be placed in positions h, h+1,..., h+r for some $1 \leq h \leq - r$. Moreover, the optimal arrangement is in a class of arrangements which we call "bell-shaped" arrangements. An arrangement is called a bell-shaped arrangement if there exists a position h ($1 \leq h \leq n$) such that $p_1/l_1 \leq \ldots \leq p_h/l_h \geq \ldots \geq p_n/l_n$. As can be seen, the arrangement is bell-shaped with respect to the p_i/l_i ratios.

Lemma 2 Let S_o be an optimal arrangement of a set of records with access probabilities $\{p_1 \ldots, p_n\}$ and the corresponding lengths $\{l_1, \ldots, l_n\}$. Let there be r + 1 records with the highest p_i/l_i ratio. Then we have

a) the records with the highest p_i/l_i ratio are placed in positions h, h+1,..., h+r in S_o for some $1 \leq h \leq n - r$,

b) $\displaystyle\sum_{i=1}^{h+r-1} p_i \geq \sum_{i=h+r+2}^{n} p_i$,

c) $\displaystyle\sum_{i=1}^{h-2} p_i \leq \sum_{i=h+1}^{n}$

and

d) $p_1/l_1 \leq p_2/l_2 \leq \ldots \leq p_h/l_n \geq p_{h+1}/l_{h+1} \geq \ldots \geq p_n/l_n$.

The results in Lemma 2 reduce greatly the number of arrangements needed to be examined in a search for optimal arrangements. Specifically, if there are r+1 records with the highest p_i/l_i ratio, then the number of arrangements needed to be examined is at most $2^{(n-r-2)}$. This number may further be reduced by effective use of b) and c) in Lemma 2. This is a significant improvement over the brute force approach where $n!/2$ arbitrary arrangements need to be examined to find the optimal arrangement. Although there appears to be a lot of structure in the optimal arrangement, the problem of finding it seems to be quite difficult. Specifically, we show that the language recognition version of this problem is NP-complete. We define the ALLOCATION PROBLEM as follows:

ALLOCATION Given a number K and a set of records with access probabilities $\{p_1,\ldots,p_n\}$ and lengths $\{l_1,\ldots,l_n\}$, is there an arrangement S of the records such that $C(S) \le K$?

We shall reduce the PARTITION problem to the ALLOCATION problem. The PARTITION problem has been shown to be NP-complete [2] and is defined as follows: PARTITION Given a set of integer $A = \{a_1,\ldots, a_n\}$, is there a partition (A_1,A_2) of the integers in A such that $\sum_{a_i \in A_1} a_i = \sum_{a_i \in A_2} a_i$?

Theorem 1 The ALLOCATION problem is NP-complete.

Proof: It is easy to see that the ALLOCATION problem is in NP.

To complete the proof, we shall show that the PARTITION problem is polynomially reducible to the ALLOCATION problem. Given an instance $A = \{a_1,\ldots a_n\}$ of the PARTITION problem, we construct an instance of the ALLOCATION problem as follows: Let there be n + 1 records with access probabilities $p_i = a_i/z$ $(1 \le i \le n)$ and $p_{n+1} = a_{n+1}/z$, where $a_{n+1} = 1 + \max_{1\le i\le n} \{a_i\}$ and $z = \sum_{i=1}^{n+1} a_i$. Let the lengths of the records be given by $l_i = a_i$ $(1 \le i \le n)$ and $l_{n+1} = x/2$, where $x = \min_{1\le i\le n} \{a_i\}$. Finally, we choose $K = K_1 + 2\lceil K_2 + K_3\rceil/z^2$, where $K_1 = [\sum_{i=1}^{n} a_i^2 + xa_{n+1}/2]/z$, $K_2 = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \sum_{m=j+1}^{n+1} a_i a_j a_m$, $K_3 = (x/2-a_{n+1})g^2$, and $g = (\sum_{i=1}^{n} a_i)/2$. It is clear that the transformation can be done in polynomial time. We omit the proof that the PARTITION problem has a solution if and only if the ALLOCATION problem has a solution. □

3. HEURISTICS

Since the ALLOCATION problem is NP-complete, we are justified in considering approximation algorithms which compute near-optimal solutions in an acceptable amount of computing time. To further motivate the importance of heuristics, we show that for any given number of records, an arbitrary arrangement can perform arbitrarily poorly in comparison with an optimal arrangement.

Theorem 2. Let S and S' be any two arrangements of a set of records with access probabilities $\{p_1, \ldots, p_n\}$ and lengths $\{l_1, \ldots, l_n\}$. Then we have

$$C(S)/C(S') \le (1 + p')/2p'),$$

where $p' = \min_{1\le i\le n} \{p_i\}$. Moreover, for every $n \ge 3$, there are sets of records for which the ratio $C(S)/C(S')$ can approach $1/(2p') + p'/2$ arbitrarily closely.

Theorem 2 shows that an arbitrary arrangement can perform quite poorly in comparison with an optimal arrangement. Since it is known from Lemma 2 that an optimal arrangement is a bell-shaped arrangement, it is interesting to see how an arbitrary bell-shaped arrangement might compare with an optimal arrangement. Before we derive this result, we need to introduce a different notation for representing bell-shaped arrangements and their expected seek cost.

We shall index the records $R_1, \ldots, R_n$ so that $p_1/l_1 \geq p_2/l_2 \geq \ldots \geq p_n/l_n$. With this indexing, any bell-shaped arrangement can be represented by an n-tuple $B = (b_1, b_2, \ldots, b_n)$, where $b_1 = 1$ and

$$b_i = \begin{cases} 1 \text{ if record } R_i \text{ is placed to the right of } R_1, \\ 0 \text{ if record } R_i \text{ is placed to the left of } R_1, \end{cases}$$

for $i = 2, \ldots, n$. Since $p_i/l_i \leq p_{i+1}/l_{i+1}$ for $1 \leq i < n$, there exist n nonnegative numbers, $d_j (1 \leq j \leq n)$, such that l_i can be expressed as $l_i = p_i \; (\sum_{j=1}^{i} d_j/p_j)$ for each $1 \leq i \leq n$. With this notation, we can rewrite the expected seek cost function as follows.

$$C(B) = \sum_{i=1}^{n} \left((d_i/p_i) \sum_{j=i}^{n} p_j(p_j + 2g_j) \right), \tag{9}$$

where $g_j = b_j \; (E(j) + V_2 \; (j)) \; V_1(j) + (1 - b_j)(E(j) + V_1(j))V_2(j)$,

$$E(j) = \sum_{k=1}^{j-1} p_k,$$

$$V_1(j) = \sum_{k=j+1}^{n} b_k \; p_k,$$

and $V_2(j) = \sum_{k=j+1}^{n} (1-b_k) \; p_k$.

Using this cost formula, we can now bound the performance of an arbitrary bell-shaped arrangement with respect to an optimal arrangement.

Theorem 3. Let B and B' be any two bell-shaped arrangements of a set of n records. Then we have

$$C(B)/C(B') \leq 1 + n/2.$$

Since an optimal arrangement is a bell-shaped arrangement, the bound in Theorem 3 also serves as a bound for the performance of an arbitrary bell-shaped arrangement with respect to an optimal arrangement. Although the bound appears to be quite large, we have been unable to find examples for which the ratio is larger than 2. Through a complicated case analysis, we have shown that the bound is in fact 2 for $n \leq 4$ and we conjecture that it holds for all n.

In addition to the bell-shaped arrangements, we have studied a couple of heuristics which are intuitively quite appealing. The first one arranges the records so that the ratios p_i/l_i form an organ-pipe arrangement. Since the arrangement obtained from this heuristic is a bell-shaped arrangement, the bound in Theorem 3 also applies to this heuristic. The second heuristic that we considered constructs two arrangements and chooses the better of the two as the final arrangement. The first is an organ-pipe arrangement by access probabilities and the second is an organ-pipe arrangement by the reciprocals of record lengths. We have shown that the worst-case performance of this heuristic with respect to optimal solutions is bounded above by $\min\{a^2/b^2, c/d\}$, where $a = \max_{1\le i\le n} \{p_i\}$, $b = \min_{1\le i\le n} \{p_i\}$, $c = \max_{1\le i\le n} \{l_i\}$, and $d = \min_{1\le i\le n} \{l_i\}$. However, we feel that this may not be a tight bound. The difficulties we encountered in deriving a tight bound for this heuristic lie in the fact that there appears to be no relationship between the two arrangements.

We should note that the two heuristics mentioned above are incomparable in the sense that there are examples in which one heuristic performs better than the other, and conversely. Surprisingly, the worst-case example we have found so far for the second heuristic has a performance ratio far less than that of the first one. We have done some experiments using randomly generated access probabilities and record lengths to get a feel for the average performance of these two heuristics. Both heuristics perform quite well in comparison with the optimal solutions, with the first heuristic outperforming the second in 19 out of 20 cases.

References

[1] Cody, R. A., "Allocation Algorithms for Minimizing Expected Access Costs on Drum-like Storage Devices," Ph.D. Thesis, Pennsylvania State University, State College, Pa., October 1974.

[2] Grossman, D. D. and Silverman, H. F., "Placement of Records on a Secondary Storage Device to Minimize Access Time," JACM, Vol. 20, No. 3, pp. 429-438, July 1973.

[3] Hardy, G. H., J. E. Littlewood and G. Polya, Inequalities, Cambridge University Press, London, 1952.

[4] Karp, R. M., "Reducibility among Combinatorial Problems," in Complexity of Computer Computation, R. E. Miller and J. W. Thatcher (eds), Plenum Press, New York, 1972, pp. 85-104.

[5] Knuth, D. E., The Art of Computer Programming: Sorting and Searching, Addison-Wesley, Reading, Massachusetts, 1973, Chapter 6.

INHERENT COMPLEXITY OF RANGE QUERY PROBLEMS

MICHAEL L. FREDMAN
University of California, San Diego
La Jolla, California

ABSTRACT

A class of on-line computational problems is defined. These problems involve two kinds of tasks: updating and retrieval. A systematic approach to the inherent complexity of implementing these tasks will be considered. We explore average complexity as well as trade-offs between the complexities of retrieval versus update, and time versus space. An application of our methods pertains to dynamic data structures which accommodate orthogonal range queries.

1. INTRODUCTION

Let S be an arbitrary commutative semi-group (set of elements closed under a commutative and associative addition operation, +), and let $V = \{v_1, v_2, \ldots, v_m\}$ be a set of variables which assume values in S. Let $T_1, T_2, \ldots, T_n$ be subsets of $\{1,2,\ldots,n\}$. We are concerned with data structures for representing V which facilitate efficient implementation of the following on-line tasks:

$$\text{Update}(j,x);\quad v_j := v_j + x,\ (x \in S,\ 1 \leq j \leq m), \tag{1}$$

$$\text{Retrieve}(j);\quad \text{return} \sum_{i \in T_j} v_i,\ (1 \leq j \leq n).$$

We define an algebraic model of computation within which these tasks are to be implemented. Relative to this computational model we present a general technique for establishing lower bounds on the inherent complexities of these tasks, including space-time trade-offs and trade-offs between the complexities of the Retrieve tasks versus the Update tasks.

A specific instance of (1) obtained by specifying m, n, and $T_1, \ldots, T_n$ shall be referred to as a computational problem. When these quantities are left unspecified, (1) defines an ensemble of such problems. Some natural choices for the commutative semi-group S are

a) the integers with the usual addition operation,

b) a family of sets with the operation of set union, and

c) the reals with a+b defined as max(a,b).

As an example illustrating choice (b), consider the following data structure problem. A set Q of records is to be represented, each record having a key in $\{1,2,\ldots,m\}$. Let v_i denote the subset of Q consisting of those records whose key value is i, and let r be a record whose key is i. Then the task Update(i,{r}) defined in (1) is tantamount to inserting r into Q. The task Retrieve(j) defined in (1) returns the subset of Q consisting of all records whose keys lie in T_j. On the other-hand, if we are only interested in determining the number of records in Q whose keys lie in some T_j, then we would choose S as in (a), and inserting r into Q would be accomplished by executing Update(i,1).

The complexity issues associated with the problems in (1) are surprisely rich. A simple example serves to illustrate this fact. Consider the problem of designing a data structure for representing an array A(j), $1 \leq j \leq n$, which stores values in S, assuming that the following tasks are to be performed.

$$\text{Update}(i,x);\quad A(i) := A(i) + x,\quad (1 \leq i \leq n \text{ and } x \in S) \qquad \text{(P)}$$

$$\text{Retrieve}(j);\quad \text{return} \sum_{i=1}^{j} A(i)\ .\quad (1 \leq j \leq n)$$

Observe that this problem corresponds to the problem in (1) obtained by specifying m=n and $T_j = \{1,2,\ldots,j\}$ for $1 \leq j \leq n$. There are three very natural data structures applicable to this problem. The most obvious data structure consists of the array A itself. With this choice, the operation Update(j,x) takes constant time (independent of n), but an execution of Retrieve can take time n in the worst case. A second approach is to maintain an array S(j), $1 \leq j \leq n$, in a manner such that the value returned by Retrieve(j) resides in S(j). With this approach, Retrieve(j) takes constant time, but an Update can take time n in the worst case. The third approach involves a minimal height binary tree with n leaves. In the jth leaf (from left to right) we store the value A(j). In each internal node, we maintain the sum of the values in the leaves of the subtree beneath that node. Update(j,x) is then executed by adding x to each node along the path from the root to the jth leaf, requiring about $\log_2 n$ additions. Retrieve(j) is executed by summing the values in an appropriate subset of nodes, also requiring about $\log_2 n$ additions in the worst case. The above data structures suggest the possibility that their may exist a family of data structures applicable to this problem, providing a range over which the time cost for executing Updates can be traded off against the cost of executing Retrieves. This problem is investigated in [1] and [4], where the following results are derived. Let C_u and C_r denote the average complexities of the Update and Retrieve tasks respectively, and let $\hat{C}_u$ and $\hat{C}_r$ denote the analogous worst case complexities. Then $\binom{\hat{C}_u + \hat{C}_r}{\hat{C}_u} \geq n+1$, and this trade-off is in fact attainable. Also $C_u = \Omega(n^{1/C_r})$ for fixed C_r, and $C_r = \Omega(n^{1/C_u})$ for fixed C_u. Finally, if we let t denote $(C_u + C_r)/2$, then $t \geq \log_4 n - O(1)$, while $t = \log_4 n + O(\log\log n)$ is attainable (as seen from consideration of worst case complexity). Observe that the binary tree approach described above only achieves $t \leq \log_2 n$, and so is non-optimal.

In the following section we define our computational model. In Section 3 we develop a general and systematic approach to complexity. Several applications of this theory are provided in Section 4.

2. COMPUTATIONAL MODEL

Let $Z = \{z_1, z_2, \ldots, z_\sigma\}$ be a set of program variables which store values in S and let $U_1, U_2, \ldots, U_m$ and $R_1, R_2, \ldots, R_n$ be finite subsets of Z. We consider algorithms which have the following straight-line format.

```
Update(i , x)
    for each $z_\ell \in U_i$ do
        $z_\ell := z_\ell + x$

Retrieve(j)
    return $\sum_{z_\ell \in R_j} z_\ell$                (2)
```

The time complexities of these programs are the set cardinalities $|U_i|$ and $|R_j|$ respectively. The worst case retrieve and update complexities are $\hat{C}_r = \max\{|R_i|\}$ and $\hat{C}_u = \max\{|U_i|\}$; similarly, the average retrieve and update complexities are respectively $C_r = \sum|R_j|/n$ and $C_u = \sum|U_i|/m$. The overall space complexity of the programs in (2)is defined as σ. We require that these programs work correctly, independently of the choice of semigroup S.

The class of programs which define this model is properly included in the class of straight-line programs. However, from the standpoint of achieving minimal complexity, it can be shown that the programs in (2) do as well as the larger class of straight-line programs.

The following lemma establishes a condition on the sets U_i and R_j equivalent to the correctness of programs in (2).

Lemma 1. The programs in (2) are semantically equivalent to the set of transactions in (1) if and only if

$$|U_i \cap R_j| = \begin{cases} 1, & \text{if } i \in T_j, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Proof: Consider an execution of the program sequence Retrieve(j), Update(i,x), Retrieve(j), and let W_1 and W_2 denote the two respective outputs. Then (1) implies that

$$W_2 = \begin{cases} W_1 + x, & \text{if } i \in T_j, \\ W_1, & \text{otherwise.} \end{cases} \tag{3a}$$

If (3) holds, then clearly (3a) holds. Because the Retrieve programs in (2) generate no side effects, it follows that the collection of programs in (2) correctly implement the transactions in (1) if (3a) holds for all i and j. Therefore, (3) is a sufficient condition for semantic equivalence.

Conversely, by choosing S to be the integers and x=1, we see that (3a) holds only if (3) holds, completing the proof.

It is worthwhile to interpret the above model and its associated complexity measures in the context of choosing S to be a family of sets of records under the union operation, continuing the discussion concerning (b) from the Introduction. Each program variable z_ℓ corresponds to a bucket of records. An execution of the task Update(i, {r}) results in the record r being placed in $|U_i|$ of these buckets. When $|U_i| > 1$, a redundant storage of records is taking place. An execution of Retrieve(j) returns a disjoint union of $|R_j|$ buckets consisting precisely of those records whose keys lie in T_j. Therefore, C_u is a measure of storage redundancy, C_r measures the number of buckets that are combined to satisfy a query, and σ represents the total number of buckets. Studying the possible trade-offs between C_u and C_r amounts to exploring the extent to which storage redundancy improves query time.

3. SYSTEMATIC APPROACH TO INHERENT COMPLEXITY

We describe a method for deriving inherent lower bounds and trade-offs involving the quantities C_u, C_r, and σ. Let Γ denote the graph having vertex set $\{u_1,\ldots,u_m, r_1,\ldots,r_n\}$ and such that an edge (u_i , r_j) is present in Γ if and only if $i \in T_j$. By Lemma 1, (u_i , r_j) is present in Γ precisely when $U_i \cap R_j \neq \phi$. No other edges are defined to be in Γ, so that Γ is a bipartite graph. We refer to Γ as the semantic graph of the computational problem in (1). Let $E(\Gamma)$ denote the set of edges in Γ. Given a complete bipartite subgraph P of Γ, let $|P|$ denote the number of vertices in P, let u_P denote the number of vertices in both P and $\{u_1,\ldots,u_m\}$, and let r_p denote the number of vertices in both P and $\{r_1,\ldots,r_n\}$. Also, let E(P) denote the set of edges of P. With each edge $e = (u_i , r_j) \in E(\Gamma)$, let w_e be an associated nonnegative number, and let w_u, w_r and w_σ denote nonnegative numbers.

Theorem 1. If $\sum_{e \in E(P)} w_e \leq u_p w_u + r_p w_r + w_\sigma$ for each complete bipartite subgraph P of Γ, then $\sum_{e \in E(\Gamma)} w_e \leq m\, w_u C_u + n\, w_r C_r + w_\sigma \sigma$.

Proof: Given a program variable $z_k \in Z$, let P_k denote the complete bipartite subgraph of Γ consisting of vertices $\{u_j \mid z_k \in U_j\} \cup \{r_j \mid z_k \in R_j\}$. Given an edge $(u_i , r_j) \in E(\Gamma)$, choose z_k to be an element in $U_i \cap R_j$. Then $(u_i , r_j) \in E(P_k)$. Therefore,

$$E(\Gamma) = \bigcup_{k=1}^{\sigma} E(P_k)$$

and consequently,

$$\sum_{e \varepsilon E(\Gamma)} w_e \leq \sum_{k=1}^{\sigma} \sum_{e \varepsilon E(P_k)} w_e \tag{4}$$

By the hypothesis of the theorem, the right hand side of (4) is dominated by

$$\sum_{k=1}^{\sigma} (u_{P_k} w_u + r_{P_k} w_r + w_\sigma). \tag{5}$$

Because $\sum_{k=1}^{\sigma} u_{P_k} = \sum_{i=1}^{m} |U_i|$ and $\sum_{k=1}^{\sigma} r_{P_k} = \sum_{i=1}^{n} |R_i|$, the sum in (5) can be rewritten as

$$m w_u C_u + n w_r C_r + w_\sigma \sigma. \tag{6}$$

Combining (4), (5), and (6) completes the proof.

Each selection of the quantities, w_u, w_r, w_σ, and w_e, $e \varepsilon E(\Gamma)$, satisfying the hypothesis of Theorem 1, determines a linear inequality involving the complexities, C_u, C_r and σ. A selection with $w_u = w_r = 0$ results in a lower bound on space, and a selection with $w_\sigma = 0$ results in a lower bound on time: more precisely, a trade-off involving C_u and C_r. The set of all inequalities derivable via Theorem 1 determines a convex region in 3-space which contains all possible values of the triple (C_u, C_r, σ).

4. APPLICATIONS

In this section we provide several applications of our method. For the purpose of defining specific instances of (1), we prefer to use a more convenient but logically equivalent notation; namely one based on the use of multiple subscripting, eg.

$$\text{Update}(i,j,x); \quad v_{ij} := v_{ij} + x, \tag{4.1}$$

$$\text{Retrieve}(k,\ell); \quad \text{Return} \sum_{(i,j) \varepsilon T_{k,\ell}} v_{ij}.$$

Our first result concerns dynamic data structures which accommodate orthogonal range queries. The purpose of such a data structure is to represent a set of records, where each record has an associated value in a commutative semi-group S, as well as an associated d-dimensional key vector over an ordered key space. An orthogonal range query is a request for the sum of the associated values of all records whose key vectors lie in a specified hypercube (cross product of d intervals). Lueker [5] gives a data structure which can perform insertions and deletions of records, as well as orthogonal range queries. This structure can be used to perform an intermixed sequence of n such operations in time $O(n(\log n)^d)$. Relative to a rather general model of computation, we can show that this result is best possible. In outline form this is argued as follows. First, a problem of the form in (1) is defined and lower bounds for this problem are derived. Secondly, a reducibility relationship between this problem and the orthogonal range query problem is derived, from which lower bounds for the latter problem are then inferred. We will discuss in detail here the first of these two steps. The second, reducibility step, is discussed in [2].

Let $[n]^d$ denote the set of n^d lattice points, $\{(i_1,\dots,i_d) \mid 1 \leq i_\ell \leq n$ for $1 \leq \ell \leq d\}$. Define

$$\text{Update}(i_1,\dots,i_d,x);\quad v_{i_1\dots i_d} := v_{i_1\dots i_d} + x, \qquad x \in S,\ (i_1,\dots,i_d) \in [n]^d \tag{7}$$

$$\text{Retrieve}(j_1,\dots,j_d);\quad \text{return} \sum_{\substack{1 \leq i_\ell \leq j_\ell \\ \text{for } 1 \leq \ell \leq d}} v_{i_1\dots i_d}, \qquad (j_1,\dots,j_d) \in [n]^d.$$

If we interpret the elements of $[n]^d$ as d-dimensional key vectors, then the Retrieve task in (7) can be seen to be one type of orthogonal range query. This is the basis for reducing the problem in (7) to the orthogonal range query problem. The problem in (7) can also be regarded as a d-dimensional variant of the problem (P) discussed in the Introduction.

Theorem 2. For fixed d, the average time complexity $t = (C_u + C_r)/2$ of the tasks in (7) satisfies $t = \Omega((\log n)^d)$.

The proof of Theorem 2 uses the following lemma.

Lemma 2. Let $a_{i_1\dots i_d}$, $i_1,\dots,i_d \geq 0$, be d-fold subscripted real variables. Then we have that

$$\sum_{\substack{i_1,\dots,i_d \geq 0 \\ j_1,\dots,j_d \geq 0}} \frac{a_{i_1\dots i_d}\, a_{j_1\dots j_d}}{(i_1+j_1+1)\dots(i_d+j_d+1)} \leq (2\pi)^d \sum_{i_1,\dots,i_d \geq 0} a^2_{i_1\dots i_d}.$$

Remark: The special case d=1 is given by Hilbert's inequality. A proof of Lemma 2 is given in [2].

Proof of Theorem 2: To simplify the exposition, we take d=2. We use Theorem 1. In keeping with our use of d-fold (d=2) subscripting, we write the vertex set of the semantic graph Γ as $\{u_{i_1 i_2}, r_{i_1 i_2} \mid (i_1, i_2) \in [n]^2\}$. An edge $(u_{i_1 i_2}, r_{j_1 j_2})$ is in $E(\Gamma)$ if and only if $i_1 \leq j_1$ and $i_2 \leq j_2$. Given an edge $e = (u_{i_1 i_2}, r_{j_1 j_2}) \in E(\Gamma)$, we define $w_e = \dfrac{c}{(j_1-i_1+1)(j_2-i_2+1)}$ where $c = 1/(2\pi)^2$, We also define $w_u = w_r = 1$ and $w_\sigma = 0$. Below we will show that the hypothesis of Theorem 1 is satisfied. Assuming this to be the case, the conclusion of Theorem 1 then reads

$$\sum_{e \,\varepsilon\, E(\Gamma)} w_e \leq 2n^2 t . \tag{8}$$

However,

$$\sum_{e \,\varepsilon\, E(\Gamma)} w_e \geq \sum_{1 \leq i_1, i_2 \leq n/2} \sum_{\substack{1 \leq k_1, k_2 \leq n/2 \\ j_1 = i_1 + k_1 - 1 \\ j_2 = i_2 + k_2 - 1}} \frac{c}{(j_1 - i_1 + 1)\,(j_2 - i_2 + 1)}$$

$$= \sum_{1 \leq i_1, i_2 \leq n/2} \sum_{1 \leq k_1, k_2 \leq n/2} \frac{c}{k_1 k_2} = c \left\lfloor \frac{n}{2} \right\rfloor^2 \left(\sum_{1 \leq k \leq n/2} \frac{1}{k} \right)^2$$

$$= \Omega(n^2 (\log n)^2) .$$

Combining this with (8) yields the conclusion in the statement of Theorem 1.

To complete the proof, we must show that the hypothesis of Theorem 1 is satisfied. This is equivalent to showing that

$$\sum_{e \,\varepsilon\, E(P)} w_e \leq (2\pi)^2 \, |P| \tag{9}$$

for any complete bipartite subgraph P of Γ.

Let $L_1 = \min\{j_1 \mid R_{j_1 j_2} \text{ is in } P\}$ and $L_2 = \min\{j_2 \mid R_{j_1 j_2} \text{ is in } P\}$.

Since P is assumed to be a complete bipartite subgraph of Γ, for each vertex $U_{i_1 i_2}$ in P, we have that $i_1 \leq L_1$ and $i_2 \leq L_2$. Given $R_{j_1 j_2}$ in P, define $n_1(j_1, j_2) = j_1 - L_1$ and $n_2(j_1, j_2) = j_2 - L_2$. Given $U_{i_1 i_2}$ in P, define $m_1(i_1, i_2) = L_1 - i_1$ and $m_2(i_1, i_2) = L_2 - i_2$. The quantities, m_1, m_2, n_1 and n_2, are nonnegative; and the sum in (9) can be written

$$\sum_{\substack{(U_{i_1 i_2}, R_{j_1 j_2}) \\ \varepsilon\, E(P)}} \frac{1}{(m_1 + n_1 + 1)\,(m_2 + n_2 + 1)} \tag{10}$$

Let $\theta = \{(m_1, m_2) \mid U_{i_1 i_2} \text{ is in } P\} \cup \{(n_1, n_2) \mid R_{j_1 j_2} \text{ is in } P\}$. Observe that $|\theta| \leq |P|$, and that θ consists of pairs of nonnegative integers. The sum in (10) does not exceed

$$\sum_{\substack{(k_1,k_2)\,\varepsilon\,\theta\\(k_1',k_2')\,\varepsilon\,\theta}} \frac{1}{(k_1+k_1'+1)\,(k_2+k_2'+1)} \tag{11}$$

Define

$$a_{k_1k_2} = \begin{cases} 1, \text{ if } (k_1,k_2)\ \varepsilon\ \theta, \\ 0, \text{ otherwise.} \end{cases}$$

Then (11) can be written

$$\sum_{k_1,k_2,k_1',k_2' \geq 0} \frac{a_{k_1k_2}a_{k_1'k'_2}}{(k_1+k_1'+1)\,(k_2+k_2'+1)}\ . \tag{12}$$

By Lemma 2, the sum in (12) does not exceed

$$(2\pi)^2 \sum_{k_1,k_2 \geq 0} a^2_{k_1k_2}\ . \tag{13}$$

By definition of $a_{k_1k_2}$, we note that the sum in (13) is simply $|\theta|$.

Because $|\theta| \leq |P|$, this sequence of inequalities establishes (9), completing the proof of Theorem 2.

In [2], an implementation of the tasks in (7) is given achieving $\hat{C}_u$, $\hat{C}_r = O((\log n)^d)$. Therefore, the lower bound of Theorem 2 is nearly best possible.

Next, we briefly describe results for two other types of range query problems. Both of these are computational problems of the type (4.1), with (i,j) and $(k,\ell)\ \varepsilon\ [n]^2$. The first problem is defined by setting

$$T_{k\ell} = \{(i,j)\ |\ (i,j)\ \varepsilon\ [n]^2 \text{ and } (i-k)^2+(j-\ell)^2 \leq n^2/16\} \tag{14}$$

Thus, Retrieve(k,ℓ) computes the sum of variables whose subscripts lie in $[n]^2$ within a radius of $n/4$ of (k,ℓ), thereby computing a "circular" range query. The second problem, which defines a "parabolic" range query, is specified by setting

$$T_{k\ell} = \{(i,j)\ \varepsilon\ [n]^2 \text{ and } j \leq \ell - (i-k)^2\}. \tag{15}$$

Let t_o and $\hat{t}_o$ denote the best (minimum) possible values for the quantities $(C_u+C_r)/2$ and max $(\hat{C}_u, \hat{C}_r)$ respectively. The problems defined in (14) and (15) are investigated in [3]. In the case of (14) it is shown that $t_0 = \Omega(\sqrt{n})$ and $\hat{t}_0 = O(n\log n)$. (These results are probably not best possible.) In the case of (15), it is shown that $t_0 = \Omega(\sqrt{n})$ and $\hat{t}_0 = O(\sqrt{n})$, a fairly tight pair of results. The $\sqrt{n}$ lower bounds are remarkably large, especially when contrasted with the $(\log n)^2$ upper bound achievable for orthogonal range queries in two dimensions.

All of the instances of (1) which we have examined satisfy m=n, and we refer to this type of problem as a balanced update retrieve problem of size n, abbr. BUR-n. Our next theorem is a general result pertaining to BUR-n's. We require the following notation.

Given the semantic graph Γ of a BUR-n, define a (k,ℓ) subgraph of Γ to be a complete bipartite subgraph with k vertices in $\{u_1,\dots,u_n\}$ and ℓ vertices in $\{r_1,\dots,r_n\}$. Define $f(j) = \max\{\ell \mid \Gamma$ contains either a (j,ℓ) subgraph or an (ℓ,j) subgraph$\}$. Observe that f(j) is a non-increasing function. Define $j_0 = \max\{\ell \mid \Gamma$ contains an (ℓ,ℓ) subgraph$\}$.

Theorem 3. For each $j \geq 1$,

$$C_u \geq \left(\frac{|E(\Gamma)|}{n} - C_r f(j)\right)/j \,, \quad \text{and} \tag{16}$$

$$C_r \geq \left(\frac{|E(\Gamma)|}{n} - C_u f(j)\right)/j \,. \tag{17}$$

Let $t = (C_u + C_r)/2$. For any b such that $0 \leq b \leq j_0$,

$$|E(\Gamma)| \leq (2bn)t + (\max\{jf(j) \mid b < j \leq j_0\})\sigma \,. \tag{18}$$

Proof: Set $w_e = 1$, $w_u = j$, $w_r = f(j)$ and $w_\sigma = 0$ in Theorem 1. To show that the hypothesis of Theorem 1 holds, observe that

$$\sum_{e\,\varepsilon\,E(P)} w_e = u_p r_p$$

If $r_p \leq j$, the $u_p r_p \leq u_p j \leq u_p w_u + r_p w_r + w_\sigma$. If $r_p > j$, then $u_p r_p - u_p w_u = u_p(r_p - j) \leq f(r_p)\,(r_p - j) \leq f(j)\,(r_p - j) \leq r_p w_r$, and so $u_p r_p \leq u_p w_u + r_p w_r + w_\sigma$. Hence, the hypothesis of Theorem 1 holds, as claimed. The conclusion of Theorem 1 establishes (16). In a similar manner, (17) is proven.

To derive (18), we set $w_e = 1$, $w_u = w_r = b$, and $w_\sigma = \max\{jf(j) \mid b < j \leq j_0\}$. We have that $\sum_{e\,\varepsilon\,E(P)} w_e = u_p r_p = jJ$ (where $j = \min(u_p, r_p)$ and $J = \max(u_p, r_p)$)

$$= b(j+J) + jJ - b(j+J) \leq b(j+J) + (j-b)J \leq b(u_p + r_p) + \begin{cases} 0, & \text{if } j \leq b, \\ jf(j), & \text{if } j > b, \end{cases}$$

$\le u_p w_u + r_p w_r + w_\sigma$. This establishes the hypothesis of Theorem 1, and (18) follows from the conclusion; completing the proof of Theorem 3.

In [1] it is shown that for each BUR-n and each k, $1 \le k \le n/\log_2 n$, there exists an implementation with $\hat{C}_u$ (respectively $\hat{C}_r$) = k, and $\hat{C}_r$ (respectively $\hat{C}_u$) $\le n/\log_2 k$. (Choosing $k = n/\log_2 n$, we conclude that $\hat{t}_0 \le n/(\log_2 n - \log_2 \log_2 n)$.) For each n there is a BUR-n such that this trade-off is approximately best possible, or in otherwords, inherent. Specifically, we have the following Corollary to Theorem 3.

Corollary 1. There exist positive constants c and d satisfying the following. For each n there exists a BUR-n such that for each $k \le cn/\log_2 n$, if C_u (respectively C_r) $\le k$, then C_r (respectively C_u) $\ge dn/(\log_2 k+1)$.

Observe that each BUR-n can be implemented in space $\sigma \le n$. Indeed, the programs in (1) which define a BUR-n use this amount of space. The next Corollary shows that for some BUR-n's, no implementation can simultaneously achieve $t = O(n/\log n)$ and $\sigma = O(n)$.

Corollary 2. Given fixed constants, c and d, for all sufficiently large n there exists a BUR-n such that no implementation can simultaneously achieve $t \le cn/\log_2 n$ and $\sigma \le dn$.

Proofs of these Corollaries appear in [1] and [4]. These proofs use Theorem 3 in conjunction with a probabilistic argument.

CONCLUSION

The computational problems in (1) in conjunction with the model in (2) provide a rich and tractable setting in which to explore the complexity of data organization. As exemplified by the results concerning the complexity of dynamic data structures which accommodate orthogonal range queries, this area of investigation can yield meaningful results concerning computational problems of genuine interest.

REFERENCES

[1] W.A. Burkhard, M.L. Fredman, and D.J. Kleitman, "Inherent complexity trade-offs for range query problems", to appear.

[2] M.L. Fredman, "A lower bound on the complexity of computing orthogonal range queries", to appear.

[3] ____________, "Lower bounds on the complexity of some optimal data structures", to appear.

[4] ____________, "A near optimal data structure for a type of range query problem", to appear.

[5] G. Lueker, "A data structure for othogonal range queries", 19th Annual Symposium on Foundations of Computer Science, 28-34, (1978).

A MULTIKEY SEARCH PROBLEM*
(Preliminary Draft)

J. IAN MUNRO
Dept. Computer Science, University of Waterloo
Waterloo, Ontario N2L 3G1, Canada.

ABSTRACT

We consider the problem of storing a file of n records each containing k keys as a simple n by k array so that records may quickly be retrieved by specifying an arbitrary subset of their keys. An upper bound of $O(n^{1-j/k})$ comparisons is established for $j < k$. A matching lower bound is shown for k=2 and stated for the general case.

1. A PROBLEM AND A SOLUTION

A sorted list is a delightful structure for performing searches based on a single key; I dare say the telephone company would be out of business without it. If comparisons are the only operations which can be performed on keys in the table, then the sorted list permits us to search as quickly as any structure which employs pointers. If basic arithmetic can be performed, the sorted list is the only structure truly amenable to interpolation search. It has, of course, its drawbacks. Modification is, at best, painful, and it is quite inappropriate if records are to be retrieved under one (or more) of several keys. Dealing with the latter problem is the main concern of this paper.

Partial match retrieval has been considered by Bentley (1), Rivest (4) and several other researchers. The point of view generally taken is that of developing a dynamic structure which employs pointers. Our attention is focussed on static structures, but those which need retain no auxiliary storage other than that required to recall the number of elements stored. The particular data structure developed is essentially an implicit representation of a structure analogous to Bentley's k-d trees (1). We show that the number of comparisons required to perform a search

---

*This work was supported by the Natural Science and Research Council of Canada under grant A8237.

on our structure is within a (small) constant factor of the minimum number required for a search over any table arranged in some fixed partial order.

The ordering can most easily be viewed as that obtained by the following construction. By a segment, we mean a sequence of consecutively stored records whose internal order has not yet been defined. Initially, the entire table is a segment. Let the keys be numbered 0..k-1.

```
Procedure Order;
Begin i := 0
   While there remain segments of more than 1 element do
   Begin
      For each segment remaining from the previous stage do
      Begin
         Move the median element under key i of the
            segment to the middle position of the segment;
         Move all elements less than this median above
            it and all elements greater, below. This
            produces 2 new segments;
      end
      i := (i+1) mod k
   end
end
```

A search on this structure can be performed by cycling through the keys. If the value of a key is known, the number of candidates is cut in half; if it is not, two subproblems of half size each remain. This leads to the result:

Theorem: A table of n records, each consisting of k keys can be arranged as an n by k array so that any record specified by j of its k keys can be found in $O(n^{1-j/k})$ comparisons if $j < k$ and $\lceil \log n \rceil$ comparisons is $j = k$. No additional information, other than the values n and k need be stored permanently with the structure.

2. THE QUESTION OF OPTIMALITY

Perhaps the most interesting aspect of this problem is the question of optimality. For simplicity we will consider only the 2 key case. First we show the above method essentially minimizes the number of comparisons required if the arrangement of the table is based on a fixed partial order. (The proof is based on a suggestion of Jai Hakim). This lower bound can be generalized to the k key case. The proof for the general case is somewhat more complicated and so will not be presented here.

Theorem: If n k-key records are arranged in a table by some fixed partial order on specific keys, and no other information is retained, then $n^{1/2}$ comparisons are necessary (in the worst case) to find a record specified by one of its keys. If both keys are specified $\lceil \log n \rceil$ comparisons are necessary.

Proof: The partial order on each key which is known from the structure can be viewed as one of two directed graphs (edge (a,b) in graph i implies key i of record a > key i of record b). The existence of (a,b) in graph i implies neither (a,b) nor (b,a) is present in the other graph.

From this we see that a chain in one graph corresponds to an independent set in the other. Since the product of the largest chain and largest independent set of any such graph must be at least n, the largest independent set in one of the graphs must be at least n. This leads immediately to the lower bound.

At a glance, we seem to have solved our problem. There remains, however, the question of whether there are better orderings which are not based on fixed partial orders. We conjecture not. Indeed we conjecture that if the n entries in a one-dimensional array are known to be in any of a set of n! possible orderings, that n comparisons are necessary (on the average) to perform a search. On the other hand, it is not unreasonable to consider schemes not based on fixed partial orderings. Consider the following construction ordering. For simplicity we stick to 2 key records, the generalization should be obvious. Think of the keys as initially sorted under the first key. We know (from Erdös and Szekeres (2)) that every sequence of n numbers contains either an increasing subsequence of length n. Find such a subsequence under the second key. We now have a sublist of length n sorted under both keys (although we are not sure whether it is sorted in increasing or decreasing order under the second. These elements (in their current order) constitute the first n elements of the final table. We return to the remaining table of n- $\sqrt{n}$ elements to determine the next n- $\sqrt{n}$ elements for the final form. It is easily seen that at most n/2 log n + O($\sqrt{n}$) comparisons are required to perform a both keys are given).

This technique is certainly not a counter example to the conjecture stated above, however, such a scheme based on a radical departure from a fixed partial order does give cause for thought. We note that in the somewhat related problem of maintaining a dictionary, (i.e. performing operations of insert delete and find) without the use of pointers the restriction for partial orders is very significant. O($\sqrt{n}$) basic operations are necessary to perform the required searches and updates if we are restricted to maintaining fixed partial orders. If however, this requirement is dropped, and the key values are known, to be distinct, O($n^{1/3}$ log n) suffice (3).

REFERENCES

(1) Bentley, J.L., "Multidimensional Binary Search Trees Used for Associative Searching", CACM 18, 9 (Sept 1975), pp. 509-516.

(2) Erdös, P. and Gy. Szekeres, "A Combinatorial Problem in Geometry", Compositio Math. 2(1935), pp. 463-470.

(3) Munro, J.I. and H. Suwanda, "Implicit Data Structures", Proc. 11th Annual ACM Symp. on Theory of Computing, May 1979, pp. 108-117.

(4) Rivest, R.L., "Partial Match Retrieval Algorithms", SIAM J. on Comp., 5,1 (March 1976), pp. 19-50.

VERY FAST INFORMATION UPDATE AND RETRIEVAL USING CELLS

BRUCE W. WEIDE
Dept. of Computer and Information Science
The Ohio State University
Columbus, Ohio 43210

ABSTRACT

We propose the use of cells to permit several information storage and retrieval operations (FIND, INSERT, DELETE, RANGE, NEAR-NEIGHBOR) to be done very rapidly. Conditions under which the operations can be done in constant expected time are developed. Part of the novelty of the cell method is that it applies to multi-dimensional data, where it also has applications to clustering of data for use with database machines.

I. INTRODUCTION

It was apparently first suggested by [IS56] and later by [KT65], among others, that certain information storage and retrieval functions (FIND, INSERT, DELETE, and even sorting) could be performed efficiently using a variant of hashing where the hashing function is sequence maintaining [W77] and distribution-dependent [K75]. Such a function H has the properties that (1) for all keys X and Y, $H(X) \leq H(Y)$ iff $X \leq Y$, and (2) each value of H is approximately equally likely to be the result of applying H to a key having an assumed input distribution. The distribution of keys need not be known in advance in order for a suitable function to be found; rather, an approximation to the empirical cumulative distribution function (ECDF) of the data usually suffices. Previous work with distribution-dependent hashing has centered on how to compute H (for example, [TK68]) and on empirical evaluations for information retrieval (FIND) applications, comparing these techniques to distribution-independent, or randomizing, hashing functions (see [DST75]). All these studies have been done for one-dimensional, or single attribute, data. In fact, [W77] argues that a direct file based on these methods is "inflexible" but "provides efficient record retrieval according to a single dimension."

Recently, [W78] and [BWY78] have given theoretical timing analyses for several different algorithms (including sorting) using the ECDF in a general divide-and-conquer framework, and for geometrical closest-point problems in particular. They have dubbed this method the cell technique. The main difference between cells and the distribution-dependent hashing of earlier work is that cells allow manipulation of multi-dimensional data.

Here, we investigate the use of cells to allow fast access to and update of files of multi-dimensional data. While the method permits constant expected-time FIND, INSERT, DELETE, UPDATE, GET-NEXT, and NEAR-NEIGHBOR queries, and very fast orthogonal RANGE queries, the constants in real databases may be large due to the parameters of current storage media. With the advent of large electron-beam addressable memories or other such fast random-access devices (see [HKK77]) the method described may seem more practical.

The structure to be maintained for each single dimension in which searching is to be done differs slightly from that normally used with hashing. It is a sorted linked list, with a linear number of cells as an auxiliary data structure allowing access to interior points of the list at approximately equally-spaced intervals (Figure 2). RANGE queries are answered in constant expected time by finding the first and last items in the range. A linked list containing all the items satisfying the query begins at the first item in the range and continues until the last item is encountered, so only the first and last items must be searched for.

A k-dimensional array of a linear number of cells is also created for each k-attribute orthogonal RANGE query that might be encountered often enough to warrant the overhead. A k-dimensional ECDF is used to directly access a cell, within which all records are linked together (Figure 3). Records are not linked between cells in these multi-dimensional versions since no natural order among data items is defined or necessary. These structures are maintained over INSERT and DELETE in time proportional to the number of such cell arrays, but independent of the number of records in the database. A RANGE query is answered in time proportional to the number of cells accessed, which should be proportional to the number of items satisfying the query if the ranges are not much smaller than their respective cell widths. In any event, the number of records to be searched exhaustively is much smaller than for other schemes that essentially examine all the records, or those within one of the ranges for intersection queries.

An important practical problem is to adapt cells to clustering of data for a database machine such as the Database Computer (DBC), which can search by hardware large buckets of data, for example, entire disk cylinders, in parallel (see [BHK79]).

The remainder of this paper is divided into two main parts. Section II discusses under what conditions good behavior can be expected from the structures to be described. Section III presents the details of the cell-based data structures being proposed.

II. THE ECDF

Many algorithms, including some for information storage and retrieval (ISR), have been analyzed under the assumption that the inputs are real numbers which are uniformly distributed between 0 and 1 (written U(0,1)). A widely known but often overlooked method is available for extending these results to arbitrary distributions, and is presented here without proof. The notation $X \sim U(a,b)$ means that X is uniformly distributed between a and b.

LEMMA 1 -
Let the random variable X have the distribution function F(x). Define the random variable Y so that $Y = F(X)$ if F is continuous at X, and $Y \sim U(F(X^-),F(X))$ if F has a jump discontinuity at X. Then $Y \sim U(0,1)$.

Intuitively, the lemma says that for a random variable X having a continuous distribution F, the random variable F(X) is uniformly distributed. The importance of this result for ISR problems rests with the two facts that (1) the transformation F is increasing, and therefore preserves the ordering of real numbers, and (2) its values are "spread out" evenly over the range [0,1]. In other words, the transformation is sequence maintaining and distribution-dependent, which are the properties required by the data structures we propose. For example, if we wish to sort a set $\{X_i\}$ of real numbers each having the distribution F, and we have an algorithm which sorts uniformly distributed real numbers, we can simply use that algorithm to sort $\{F(X_i)\}$, assuming F is continuous. The analysis of the algorithm for uniformly distributed inputs will still apply, since $F(X_i) \sim U(0,1)$. Furthermore, the set $\{X_i\}$ will be sorted when the set $\{F(X_i)\}$ is sorted because $F(X_i) \leq F(X_j)$ iff $X_i \leq X_j$. The minor modification which seems to be required if F is not continuous may not really be necessary if our sorting algorithm is sophisticated enough to recognize equal items as a special case. In any event, it is easy enough to implement the transformation in constant time per item, adding only $O(n)$ steps to the time used by the algorithm.

This approach is fine if F is known and if we can compute $F(x)$. In many practical circumstances, however, F is not given as part of the problem. Hereafter, we assume the data are independent identically distributed random variables having the (unknown) distribution F. One obvious solution, of course, is to try to estimate F by sampling the inputs, and to produce a function called the empirical cumulative distribution function (ECDF) which is close to F. Given a random sample $\{X_i\}$ of size m from a distribution $F(x)$, let $X_{(j)}$ be the j^{th}-smallest sample value. We define the ECDF $F_m(x)$ as

$$\begin{aligned} F_m(x) &= 0 & x &< X_{(1)} \\ &= (j-1+(x - X_{(j)})/(X_{(j+1)} - X_{(j)})) \,/\, (m-1), & X_{(j)} &\leq x \leq X_{(j+1)} \\ &= 1 & X_{(m)} &\leq x \end{aligned}$$

This definition of the ECDF is not standard, but has special properties which will be of particular interest to us. The usual ECDF, call it $G_m(x)$, has $G_m(x) = j/m$ for $X_{(j)} \leq x < X_{(j+1)}$ and $1 \leq j < m$. Figure 1 shows what F_m and G_m might look like for $m = 4$. Notice that F_m is continuous, whereas G_m is not. From now on, we will consider m to be a constant which is small in practice.

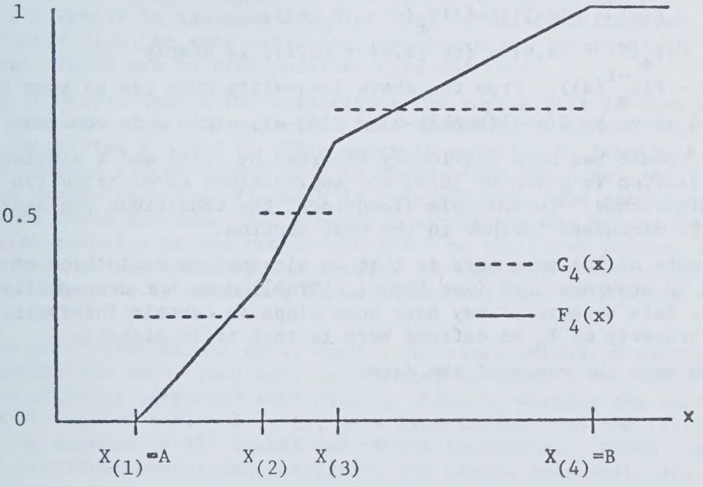


FIGURE 1 - The Empirical Cumulative Distribution Function

Under what circumstances is F_m an adequate substitute for F? It is clear that F_m is increasing, and is therefore sequence maintaining. However, it should also have the property that the result of applying it to data from the underlying distribution F is approximately uniformly distributed. These are the characteristics required in order for our data structures to permit efficient ISR operations.

The specific property corresponding to the second characteristic (approximate uniformity of the result of applying F_m) is that there exists a constant C such that for every interval $[a,b] \in [0,1]$ we have $P\{F_m(X) \in [a,b]\} \leq C(b-a)$ whenever X has the distribution F. If this condition is satisfied, we say that F_m spreads the data. Furthermore, we say that F satisfies condition L (similar to but slightly more restrictive than a Lipschitz condition) iff (1) there exist finite constants A,B such that $F(x) = 0$ for $x \leq A$ and $F(x) = 1$ for $x \geq B$, and (2) there exists a constant C such that for every x and y satisfying $A \leq x \leq y \leq B$, $F(y) - F(x) \leq C(y-x)$. This condition is tantamount to saying that F is the distribution function for a bounded random variable, and has a density that is everywhere bounded above.

The main result characterizing conditions under which F_m can adequately substitute for F is:

LEMMA 2 -
If F satisfies condition L, then F_m spreads the data.

Proof: Let A,B,C be the constants whose existence is assured because F satisfies condition L. Then for $A \leq x \leq y \leq B$, we have $F(y) - F(x) \leq C(y-x)$. From the construction of F_m we know that

$$(F_m(y) - F_m(x))/(y-x) \geq 1/(m-1)(B-A),$$

the right hand side being the minimum possible slope of F_m between A and B. Combining these inequalities gives:

$$F(y) - F(x) \leq C(m-1)(B-A)(F_m(y) - F_m(x)).$$

Now, $P\{F_m(X) \in [a,b]\}$, for $[a,b] \in [0,1]$, is simply $F(F_m^{-1}(b)) - F(F_m^{-1}(a))$. From the above inequality this can be seen to be bounded above by $C(m-1)(B-A)(b-a) = C'(b-a)$, since m is constant. ☐

This result has been previously reported by [W78] and a similar characterization is given by [D79] for applications to distributive sorting algorithms. In multiple dimensions, the conditions are analogous, and will be discussed further in the next section.

One note of interest here is that an alternative definition of the ECDF used in previous work (see [IS56], [KT65]) does not necessarily spread the data because it may have zero slope in certain intervals. The nice property of F_m as defined here is that it is strictly increasing over the range of the data.

III. DATA STRUCTURES USING THE ECDF

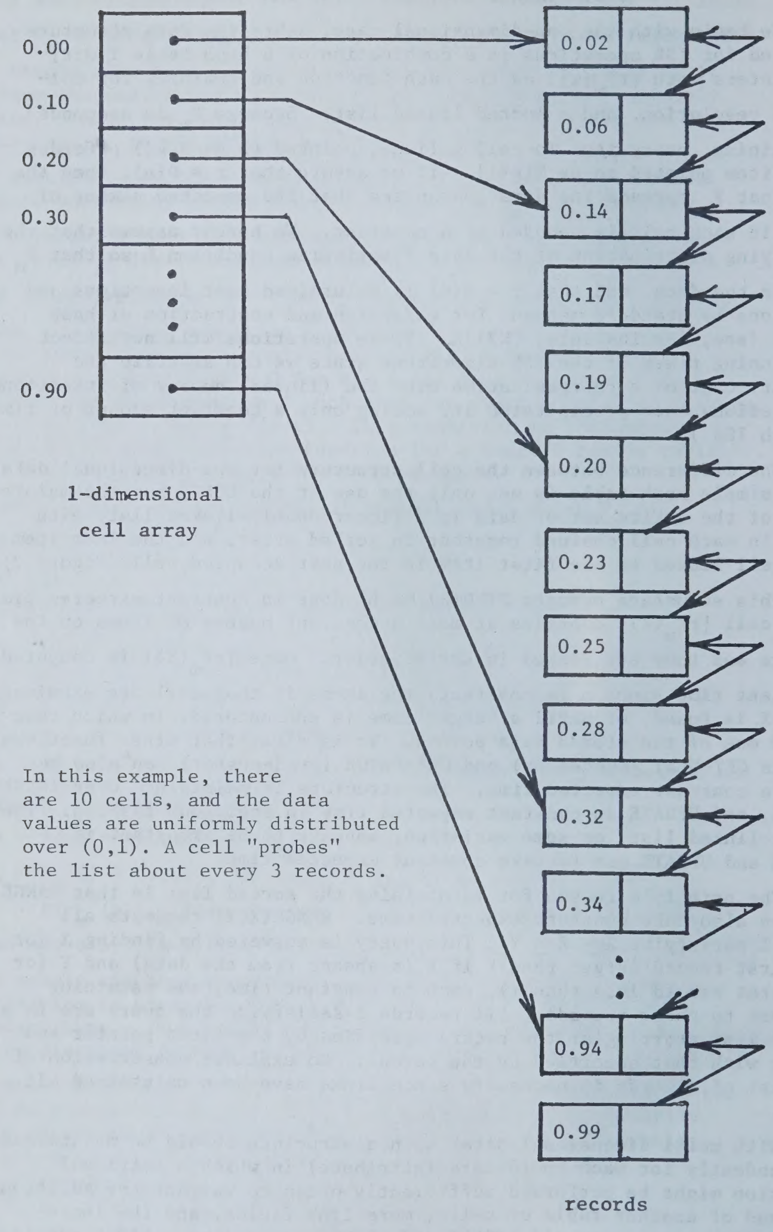
We begin with the one-dimensional case, wehre the data structure proposed for ISR operations is a combination of a hash table T[0:r] of pointers with $\lfloor rF_m(x) \rfloor$ as the hash function and chaining for collision resolution, and a sorted linked list. Because F_m is sequence maintaining, every item in cell k (i.e., pointed to by T[k]) precedes every item pointed to by T[k+1]. If we assure that $r = \theta(n)$, then the fact that F_m spreads the data guarantees that the expected number of items in each cell is bounded by a constant. We hereby assume that the underlying distribution of the data F satisfies condition L so that F_m spreads the data, and that $r = \theta(n)$ is maintained over insertions and deletions by standard methods for expansion and contraction of hash tables (see, for instance, [K71]). (These operations will not affect the running times of the ISR algorithms since we can amortize the (linear) cost of a reorganization over the (linear) number of insertions or deletions that precipitated it, adding only a constant amount of time to each ISR.)

The difference between the cell structure for one-dimensional data and a simple hash table is not only the use of the ECDF, but the maintenance of the entire set of data in a linear doubly-linked list, with items in each cell chained together in sorted order, and the last item in a cell linked to the first item in the next occupied cell (Figure 2).

This structure permits FIND(X) to be done in constant expected time, since cell $\lfloor rF_m(X) \rfloor$ contains at most a constant number of items on the average and they are linked in sorted order. Once $\lfloor rF_m(X) \rfloor$ is computed (constant time since m is constant) the items in that cell are examined until X is found, or until a larger itme is encountered, in which case X is not one of the stored data points. It is clear that other functions such as GET-NEXT (successor) and GET-PRIOR (predecessor) can also be done in constant expected time. The structure is maintained over INSERT, DELETE, and UPDATE in constant expected time in analogous fashion. The doubly-linked list, or some variation, appears to be important if DELETE and UPDATE are to take constant expected time.

The principle reason for maintaining the sorted list is that RANGE queries also take constant expected time. RANGE(X,Y) requests all items Z satisfying $X \leq Z \leq Y$. This query is answered by finding X (or the first record larger than X if X is absent from the data) and Y (or the first record less than Y), each in constant time, and returning pointers to those records. All records Z satisfying the query are in a linked list starting at the record specified by the first pointer and ending with that specified by the second. No explicit construction of the list of records is necessary since links have been maintained all along.

With multi-dimensional data, such a structure should be maintained independently for each coordinate (attribute) in which a retrieval operation might be performed sufficiently often to warrant the additional overhead of another table of cells, more link fields, and the incremental time expense to all INSERT and DELETE operations. RANGE queries or other retrieval operations specifying any single coordinate are conducted as before.


0.00
0.10
0.20
0.30
0.90
1-dimensional
cell array
0.02
0.06
0.14
0.17
0.19
0.20
0.23
0.25
0.28
0.32
0.34
0.94
0.99
records

In this example, there are 10 cells, and the data values are uniformly distributed over (0,1). A cell "probes" the list about every 3 records.

FIGURE 2 - One-Dimensional Data

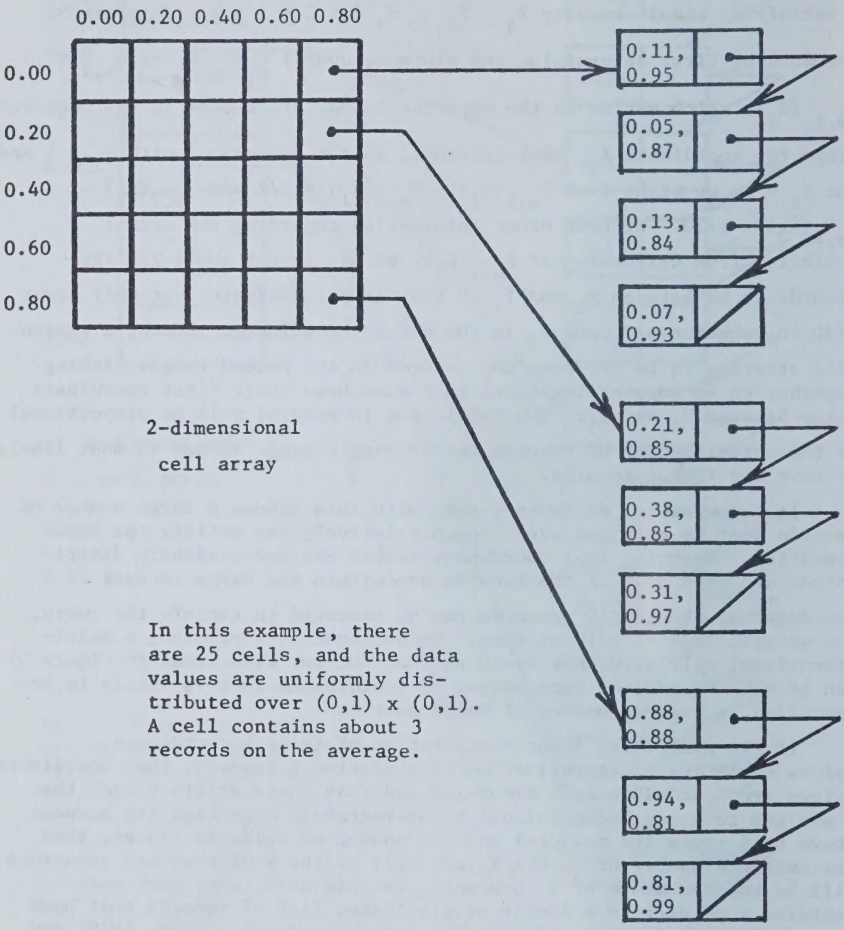
Even orthogonal RANGE queries, or intersection queries, are facilitated by this collection of one-dimensional structures. Such a query, RANGE$((i_1, X_1, Y_1), \ldots (i_k, X_k, Y_k))$, asks for all records $\vec{Z}$ satisfying simultaneously $X_j \leq Z_{i_j} \leq Y_j$ for $1 \leq j \leq k$. This can be answered by first determining the minimum, over $1 \leq j \leq k$, of $F_{m,i_j}(Y_j) - F_{m,i_j}(X_j)$, which estimates the expected number of records in the required range for coordinate i_j. For instance, in two dimensions with $i_1 = 1$ and and $i_2 = 2$, we might find $F_{m,1}(Y_1) - F_{m,1}(X_1) = 1/2$ and $F_{m,2}(Y_2) - F_{m,2}(X_2) = 1/30$. Without other information regarding the actual deviation from uniformity of $F_{m,i}(X_i)$, we may expect about 1/2 the records to be between X_1 and Y_1 in the first coordinate, but only about 1/30 to be between X_2 and Y_2 in the second coordinate. Hence, a reasonable strategy is to traverse the records in the second range, linking together on an answer list those that also have their first coordinate value between X_1 and Y_1. The total time in general will be proportional to the actual number of records in the single range chosen as most likely to have the fewest records.

It is possible, of course, that with this scheme a large number of records must be examined even though relatively few satisfy the RANGE condition. Assuming that coordinate values are independently distributed, and that 1/10 of the records are within the range in each of k coordinates, about $n/10^k$ records can be expected to satisfy the query, but we will look at n/10 of them. To overcome this problem, a multi-dimensional cell structure (such as that for two dimensions in Figure 3) can be constructed for each subset of coordinates that is likely to be specified in a large number of RANGE queries.

If the underlying joint distribution of those k coordinate values satisfies a generalization of condition L (namely, that coordinate values are bounded in each dimension and that there exists C such that F assigns to every k-dimensional hyper-rectangle a probability bounded above by C times its measure) and the number of cells is linear, then the expected number of points in any cell of the k-dimensional structure will be bounded above by a constant. In this structure, each cell contains a pointer to a linear singly-linked list of records that hash there under a k-dimensional ECDF for those coordinates. See [W78] and [BWY78] for a discussion of some alternatives for constructing a suitable ECDF in k-dimensions, and concerning the choice of the sample size m.

A RANGE search in this structure is done by identifying those cells that potentially contain records in the desired range and then simply searching those cells. For instance, in Figure 3, if we want records with both coordinate values between 0.85 and 0.95, we search only the lower right cell; the alternative method calls for examining all records with, say, the first coordinate between 0.85 and 0.95. With this example, the two-dimensional structure could be expected to save a factor of about 2.5 in the time required to answer this query.

[W78] and [BWY78] have shown how to use this structure to find nearest neighbors and to solve other closest-point problems rapidly. Their analysis assumes that the underlying distribution F satisfies a


0.00 0.20 0.40 0.60 0.80
0.00
0.20
0.40
0.60
0.80
0.11, 0.95
0.05, 0.87
0.13, 0.84
0.07, 0.93
0.21, 0.85
0.38, 0.85
0.31, 0.97
0.88, 0.88
0.94, 0.81
0.81, 0.99
2-dimensional cell array
records

In this example, there are 25 cells, and the data values are uniformly distributed over (0,1) x (0,1). A cell contains about 3 records on the average.

FIGURE 3 - Two-Dimensional Data

restricted condition L (call it condition LA) where the expected number of points per cell is bounded above and is bounded below away from zero. If condition LA is satisfied, and the range in question entirely contains at least one cell, then the expected number of records examined in a RANGE query is proportional to the expected number of records that satisfy it. This will actually be true so long as no range in any coordinate is substantially smaller than the corresponding cell width in that dimension. If this is not true, though, then there will be one dimension for which very few records are in the required range, and the brute-force search of those records will be fast anyway. In a sense, then, the multi-dimensional cell structure offers protection against RANGE queries satisfied by many records, while the single-dimensional cell structures can be used to answer those satisfied by relatively few records. The determination of which strategy to use can be made in constant time by first examining the individual coordinates, and if too many records are involved, switching to the multi-dimensional structure.

IV. CONCLUSIONS

The purpose of this paper has been to resurrect interest in a practical idea that first appeared at least 20 years ago, but has been almost ignored since. The use of the ECDF as a hashing function, combined with the maintenance of data items in a sorted list, facilitates information storage and retrieval operations. In fact, under rather weak conditions on the underlying distribution of the data, the operations take constant expected time. An auxiliary multi-dimensional cell structure aids in answering orthogonal RANGE queries without detracting from the running times of the other operations.

Cells have also been proposed for closest-point problems ([W78], [BWY78]), and Jon Bentley reports that the expected number of points per cell should be between about 1 and 10 to minimize actual running time for an implementation of NEAREST-NEIGHBOR searching. More studies of such practical considerations remain to be done. For instance, how to choose the sample size m, how to best construct and compute the multi-dimensional ECDF, and how to apply cells to clustering of data for use in database machines (see [BHK79]) are among the questions still under investigation.

ACKNOWLEDGEMENTS

The author is indebted to Jon Bentley and Michael Shamos for their continuing interest in and help with the problems and solutions presented here.

REFERENCES

1. [BHK79] Banerjee, J., Hsiao, D. K., and Kannan, K. DBC - A database computer for very large databases. IEEE Trans. on Comp. C-28, 6 (June 1979), 414-429.
2. [BWY78] Bentley, J. L., Weide, B. W., and Yao, A. C. Optimal expected-time algorithms for closest-point problems. Proc. 16th Annual Allerton Conf. on Comm., Contr., and Computing, University of Illinois, Oct. 3-5, 1978, pp. 843-851.
3. [D79] Devroye, L. Average Time Behavior of Distributive Sorting Algorithms. School of Comp. Sci., McGill Univ., Montreal, PQ, Mar. 1979, T.R. No. SOCS 79.4.
4. [DST75] Deutscher, R. F., Sorenson, P. G., and Tremblay, J. P. Distribution-dependent hashing functions and their characteristics. Proc. 1975 SIGMOD Conf., ACM, 1975, pp. 224-236.
5. [HKK77] Hsiao, D. K., Kannan, K., and Kerr, D. S. Structure memory designs for a database computer. Proc. 1977 Annual Conf. of ACM, ACM, 1977, pp. 343-350.
6. [IS56] Isaac, E. J., and Singleton, R. C. Sorting by address calculation. JACM 3, 3 (July 1956), 169-174.
7. [K71] Knott, G. D. Expandable open addressing hash table storage and retrieval. Proc. SIGFIDET Workshop on Data Descr., Access, Control, ACM, 1971, pp. 186-206.
8. [K75] Knott, G. D. Hashing functions. Comp. J. 18, 3 (Aug. 1975), 265-278.
9. [KT65] Kronmal, R. A. and Tarter, M. E. Cumulative polygon address calculation sorting. Proc. 20th National ACM Conf., ACM, 1965, pp. 376-384.
10. [TK68] Tarter, M. E., and Kronmal, R. A. Estimation of the cumulative by Fourier series methods and application to the insertion problem. Proc. 23rd National ACM Conf., ACM, 1968, pp. 491-497.
11. [W78] Weide, B. W. Statistical Methods in Algorithm Design and Analysis. Dept. of Comp. Sci., Carnegie-Mellon Univ., Pittsburgh, PA., Aug. 1978, CMU-CS-78-142.
12. [W77] Wiederhold, G. Database Design. McGraw-Hill, N.Y., 1977.

IMPROVING STORAGE UTILIZATION IN BALANCED TREES[†] (Extended Abstract)

GREG N. FREDERICKSON
Department of Computer Science, The Pennsylvania State University
University Park, PA 16802

Abstract

Recent analyses [5, 6] of random balanced trees indicate an asymptotic storage utilization rate of $\ln 2 \approx 69.3\%$. We consider enhancing the storage utilization rate by allowing more than one size of node in the tree. We extend the analysis to handle these cases, and show that, e.g., two node sizes allow an increase in the utilization rate to 84%. We also analyze a modification of B*-trees [2, 4], and exhibit an asymptotic storage utilization rate of 83.2%.

1. Introduction

A balanced tree structure, called a B-tree and introduced by Bayer and McCreight [2], is widely used in information systems. The scheme is used for storing large files on mass storage, normally with a node in the tree corresponding to a page on a disk. Each node has between $\lfloor (m-1)/2 \rfloor$ and $m-1$ keys and associated records, where m is the order of the B-tree. The tree is balanced in the sense that all leaves are on the same level. Insertions of new keys are done at the leaves. If insertion into a node would result in a node with m keys, the keys are divided between two nodes of $\lfloor (m-1)/2 \rfloor$ and $\lceil (m-1)/2 \rceil$ keys, and the remaining key is inserted into the parent node. A slightly different scheme uses the interior nodes as an index on the keys, and maintains all records in the leaves. This structure, which we call a B'-tree, is a generalization of Hopcroft's 2-3 tree as presented in [1].

While balanced trees of the above variety are designed mainly to minimize disk accesses, it is reasonable to consider the average amount of space required, since in worst case every node may be only half filled. Yao [6] has analyzed the storage utilization in random B-trees. His analysis considers the utilization of nodes in the lowest levels of B-trees of order 3 and higher orders, and shows that the utilization rate declines to $\ln 2 \approx 69.3\%$ for B-trees of sufficiently large order. Nakamura and Mizoguchi [5] have presented a similar analysis for a block storage scheme that may be viewed as the lowest level of a B'-tree. They achieve the same asymptotic utilization rate of 69.3%. These rates are in the form of the ratio of the number of records to the expected number of record positions.

In their original paper [2], Bayer and McCreight express reservations about the potentially low storage utilization rate in B-trees. They propose another structure, called a B*-tree in [4], in which an insertion that causes overflow in one node is rotated to a brother node. When the brother node is also full, the two nodes are split into three nodes, each approximately two thirds full. While their method enhances storage utilization, it appears that an analysis along the lines of that in [6] is not possible.

We consider a modification of B*-trees, which we call B^+-trees, for which we can perform analysis. We group nodes into sets of either two or three brother nodes that are used to absorb overflow insertions. Hence our method appears to be midway in performance between the two alternative implementations of B*-trees mentioned in [4]: checking one brother node upon overflow, and checking (when possible) both left and right brother nodes. With our

---

[†] Work supported in part by the National Science Foundation under Grant MCS-7909259.

modification, we are able to achieve, for sufficiently many keys, an asymptotic rate of 83.2%.

We observe that B^+-trees are essentially a special case of an alternative method of increasing storage utilization, which we also investigate. The alternative strategy involves using nodes of more than one size. With two node sizes, $m_1 > m_2$, a node with i records is stored in a page of size m_2, if $\lfloor (m_1 - 1)/2 \rfloor \le i \le m_2 - 1$, and in a page of size m_1 if $m_2 \le i \le m_1 - 1$. The scheme is a simplification of an idea proposed in [NM], for which no analysis was supplied. Of course, it is easy to see that the worst-case utilization rate is increased to approximately $1/\sqrt{2} \approx 70.7\%$ when m_2 is chosen to be $m_1/\sqrt{2}$.

We extend the methods of [5] and [6] to analyze the utilization rate in balanced trees with more than one node size. We develop formulas in terms of m_1 for the ratio of the expected number of records to the expected number of record positions in the lowest level of B- and B'-trees. For sufficiently large random trees and node sizes, we achieve $(1 + \sqrt{2})(\ln 2)/2 \approx 84\%$ for two node sizes in optimal ratio. In general, for $r \ge 2$ different node sizes we achieve a rate of $(\ln 2)/(r(2^{1/r} - 1))$. Our method thus identifies a trade-off (in constant factors) between storage utilization and overhead to handle insertions.

In section 2 we define the balanced trees we analyze, and consider the probability assumptions under which we analyze them. In section 3 we review the analysis of [5] and [6]. In section 4 we extend the results of section 3 to the case in which there is more than one node size. In section 5, we introduce and analyze a modified version of the Bayer and McCreight overflow technique.

2. Definitions

We use the definition of a B-tree as found in [4]. A B-tree of order m is a tree satisfying the following properties:

1) Every node has at most m sons.
2) Every node, except for the root and the leaves, has at least $\lfloor (m+1)/2 \rfloor$ sons.
3) The root has at least 2 sons (unless it is a leaf).
4) All leaves appear on the same level and carry no information.
5) A nonleaf node with i sons contains $i-1$ keys.

Since the leaves carry no information, we may regard them as external to the tree, with pointers to them being null.

A B'-tree of order m is a tree satisfying properties 1) - 3) from above plus the following properties:

4') All leaves appear on the same level and carry information.
5') A non-leaf node with i sons contains $i-1$ keys, where the jth key is the largest value in the subtree rooted at the jth son.

The keys in the internal nodes serve as indices to complete records in the leaves. We consider a variant of the above, in which the records in the leaves are moved up one level. Thus the records are packed in sequential order into blocks, which occupy the lowest level of the tree. Nodes containing index keys occupy all levels of the tree above the lowest level. We note that, with the exception of the extreme right block, a record cannot be inserted to the right of the rightmost key in a block.

An n-key random B-tree is a B-tree generated by inserting n keys, $k_1, k_2, \ldots, k_n$, into an initially empty tree, assuming that each of the $n!$ permutations of the keys are equally likely. This is the definition used by Yao [6], and is, as he notes, equivalent to a tree obtained by n random insertions. It is easy to see that insertion into each insertion position (a null pointer) is equally likely.

An n-key almost-random B'-tree is a B'-tree generated by inserting n keys into an initially empty tree, where the $(n-1)!$ permutations of the keys with the largest key coming first are equally likely. The largest key will occupy the rightmost position in the rightmost block of the tree, and no keys will be inserted to the right of it. Hence this assumption is made to force all nodes at the lowest level to have similar insertion characteristics: A block with i keys will have i insertion positions, each of which is equally likely. The immediately preceding condition is the assumption made by Nakamura and Mizoguchi [5], without complete justification. Our assumption about the largest key may be enforced by inserting a dummy key of very large size into the tree. As the number n of keys becomes large, the effect of this dummy key becomes negligible.

3. Expected storage utilization for lowest level of B- and B'-trees

We now lay the groundwork for the analysis for both types of trees. Let $A_i(n)$ be the expected number of nodes with i records in the lowest level of an n-key random B-tree, where $\lfloor (m-1)/2 \rfloor \le i \le m-1$. Let $A'_i(n)$ be the corresponding quantity for an almost-random B'-tree, where $\lfloor m/2 \rfloor \le i \le m-1$. The expected number of insertion positions in nodes with i records in a B-tree is $(i+1)\,A_i(n)$. Since the total number of insertion positions is $n+1$, the probability of inserting into a node with i records is $(i+1)\,A_i(n)/(n+1)$. The corresponding probability for a B'-tree is $i\,A_i(n)/n$, since each node with i records will have i insertion positions. Upon insertion into a node with $m-1$ records in a B-tree, one record is inserted into the parent node, and two nodes of size $\lfloor (m-1)/2 \rfloor$ and $\lceil (m-1)/2 \rceil$ are created. The corresponding insertion in a B'-tree does not result in a record being moved to the parent node (although a key may be inserted). Hence two nodes of size $\lfloor m/2 \rfloor$ and $\lceil m/2 \rceil$ are created.

We may then give four sets of recurrence equations, a set for each type of tree and parity (odd or even) of m. We refer the reader to [Y] for B-trees, and we consider, for simplicity, the case of $m = 2p$ for B'-trees. Hence we get:

$$A'_i(n+1) = A'_i(n) + ((i-1)A'_{i-1}(n) - iA'_i(n))/n \quad \text{for} \quad p < i \le m-1$$

$$A'_p(n+1) = A'_p(n) + (2(m-1)\,A'_{m-1}(n) - pA'_p(n))/n \tag{1}$$

We may describe the above set of equations in the matrix notation:

$$A'(n+1) = (I' + B'/n)\,A'(n) \tag{2}$$

where $A'(n) = [A'_p(n)\ A'_{p+1}(n)\ \ldots\ A'_{m-1}(n)]^T$, I' is a $p \times p$ identity matrix, and B' is

$$B' = \begin{bmatrix} -p & & & & 2(m-1) \\ p & -(p+1) & & & 0 \\ & +(p+1) & -(p+2) & & 0 \\ & & \ddots & \ddots & \vdots \\ & & & \ddots \quad \ddots & 0 \\ & & & m-2 & -(m-1) \end{bmatrix}$$

We now introduce notation for the limit of $A'(n)$, as n gets large.

$$A' = \lim_{n \to \infty} A'(n)/n$$

Taking the limit of equation (2), and performing some arithmetic, we get:

$$A' = B'A' \tag{3}$$

We may introduce similar notation for B-trees:

$$A = \lim_{n \to \infty} A(n)/(n+1)$$

We note that A_i and A'_i are the asymptotic probabilities that a record is in a node with i records. We solve equation (3) below, but omit arguments of convergence, and refer the reader to [6].

Lemma 1. Let A_i and A'_i be the asymptotic probabilities that records in the lowest level of a random B-tree and B'-tree, respectively, are in nodes with i records. Then

$$A_i = \begin{cases} \dfrac{(m+1)m}{(i+2)(i+1)} A_{m-1} & \lceil \frac{m-1}{2} \rceil \le i \le m-1 \\[2ex] \dfrac{m}{\lceil \frac{m-1}{2} \rceil + 1} & i = \lfloor \frac{m-1}{2} \rfloor \ne \lceil \frac{m-1}{2} \rceil \end{cases}$$

$$A'_i = \begin{cases} \dfrac{m(m-1)}{(i+1)i} A'_{m-1} & \lceil \frac{m}{2} \rceil \le i \le m-1 \\[2ex] \dfrac{(m-1)}{\lceil \frac{m}{2} \rceil} A'_{m-1} & i = \lfloor \frac{m}{2} \rfloor \ne \lceil \frac{m}{2} \rceil \end{cases}$$

Proof. From the solution of equation (3) for B'-trees with $m = 2p$, and similar equations for other cases. □

From summing equation (3), we get

$$\sum_{i=p}^{m-1} A'_i = (m-1)A'_{m-1} \tag{4}$$

Since all records in a B'-tree are on the lowest level, we have, in the limit:

$$\sum_{i=p}^{m-1} iA'_i = 1. \qquad (5)$$

We define the following functions:

$$H(k) = \sum_{i=1}^{k} \frac{1}{i} \text{ for } k \geq 1,$$

$$G(m) = \begin{cases} H(m) - H(m/2) & \text{if } m \text{ is even} \\ H(m+1) - H((m+1)/2) & \text{if } m \text{ is odd} \end{cases}$$

From [3, p. 72] we know that $H(m) \sim \ln m + 0.58 + 1/2m + \ldots$. Hence $G(m) \sim \ln 2 + O(1/m)$.

Lemma 2. Let A_{m-1} and A'_{m-1} be the asymptotic probabilities that records in the lowest level of random B- and B'-trees, respectively, are in nodes with $m - 1$ records. Then

$$1/A_{m-1} = (m+1)m\, G(m)$$

$$1/A'_{m-1} = m(m-1)\, G(m-1).$$

Proof. By substituting the results of Lemma 1 into equation (5) for B'-trees with $m = 2p$, and similar equations for other cases. ☐

Let $U_m(n)$ be the ratio of the expected number of records to the expected number of record positions in the lowest level of an n-key random B-tree of order m. Let $U'_m(n)$ be the corresponding quantity in an n-key almost-random B'-tree. Let U_m and U'_m be the corresponding limits, as n approaches ∞. We now state the theorem inherent in the work of [5] and [6].

Theorem 1. Let the asymptotic storage utilization rate U_m and U' for B- and B'-trees of order m, respectively, be defined as above. Then

$$U_m = \frac{1}{m-1}\,[(m+1)\,G(m) - 1]$$

$$U_m = \frac{1}{m-1}\,[m\,G(m-1].$$

Proof. For B'-trees with $m = 2p$, we take the limit of $U'_m(n)$ as n grows large:

$$U'_m = 1/[(m-1) \sum_{i=p}^{m-1} A'_i]$$

Substituting using equation (4) and Lemma 2 yields the desired result. Other results derive similarly. ☐

Corollary 1. Let U_m and U'_m be as in Theorem 1. Then

$$\lim_{m \to \infty} U_m = \lim_{m \to \infty} U'_m = \ln 2.$$

4. Storage utilization with more than one page size

We consider storing records in nodes of $r \geq 2$ different sizes $m_1, m_2, \ldots, m_r$, where a node of size m_i has at most m_i sons and $m_i - 1$ records, and $m_1 > \ldots > m_r \geq \lceil m_1/2 \rceil - 1$. A B-tree whose node sizes satisfy the above properties is called a B-tree of order M, where $M = \{m_1, m_2, \ldots, m_r\}$. A similar definition applies to B'-trees of order M. Insertion into a node with $m_i - 1$ records, $1 < i \leq r$, forces the node to be copied into a node with $m_{i-1} - 1$ record positions. Hence the additional overhead associated with insertions.

We are interested in the values of U_M and U'_M, the ratios of the expected number of records to the expected number of record positions in the lowest level of B- and B'-trees. We derive values for U_M and U'_M and determine maximum values for U_M and U'_M as a function of m_1.

Theorem 2. Let U_M and U'_M, as defined above, be the asymptotic storage utilization rates for B- and B'-trees of order M, where $M = \{m_1, m_2, \ldots, m_r\}$. Then

$$U_M = ((m_1 + 1)\ G(m_1) - 1)/D_M$$

$$U'_M = m_1\ G(m_1 - 1)/D'_M$$

where

$$D_M = 2m_r + (m_1 + 1)(-r + \sum_{j=2}^{r} (m_{j-1} + 1)/(m_j + 1))$$

$$D'_M = 2m_r - 1 + m_1\ (-r + \sum_{j=2}^{r} m_{j-1}/m_j).$$

Proof. From the definition of $U'_M(n)$, we derive:

$$U'_M = 1/[(m_r - 1) \sum_{i=m_r+1}^{m_1-1} A'_i + \sum_{j=1}^{r-1} (m_j - m_{j+1}) \sum_{i=m_{j+1}}^{m_1-1} A'_i].$$

By summing selected equations from (3) and substituting from Lemma 1, we get:

$$\sum_{i=m_j}^{m_1-1} A'_i = (m_1/m_j - 1)(m_1 - 1)\ A'_{m_1-1}.$$

Substituting into the expression for U'_M, and then substituting for A'_{m_1-1} yields the desired result. The results for B-trees derive similarly.

We note that the expression of Theorem 2 agrees with that of Theorem 1 in the case that $r = 1$. For $r = 2$ node sizes, a convenient ratio appears to be $2m_1 = 3m_2$. The nodes may then be allocated out of a page of size

$2m_1$. Using the result from Theorem 2, we find the asymptotic utilization rate to be $(6/5) \ln 2 \approx 83.2\%$. We generalize this observation in the following result. We define the function

$$\tilde{G}(2s - 1) = H(2s - 1) - H(s - 1)$$

We note that as s becomes large, $\tilde{G}(2s - 1)$ approaches $\ln 2$.

Corollary 2. The storage utilization rates for B-trees of order $M = \{m_1, m_2, \ldots, m_r\}$, where $r(m_1 + 1) = (r - 1 + i)(m_i + 1)$ for $1 \leq i \leq r$, and B'-trees where $rm_1 = (r - 1 + i)m_i$, approach:

$$U_M = ((m_1 + 1)\, G(m_1) - 1)/((m_1 + 1)\, \tilde{G}(2r - 1) - 2)$$

$$U'_M = m_1\, G(m_1 - 1)/(m_1\, \tilde{G}(2r - 1) - 1).$$

We next consider the optimal utilization rate that can be achieved for any specific $r \geq 2$ and m_1. We determine the optimal values of m_i for $1 < i \leq r$, ignoring the requirement that the m_i's be integers. We hence derive an upper bound on the utilization rate for any r and m_1.

Theorem 3. For fixed r, the expression for the utilization rate U_M achieves a maximum for

$$m_i + 1 = 2^{-\frac{i-1}{r}} (m_1 + 1) \quad \text{for} \quad 1 < i \leq r$$

The rate U'_r achieves a maximum for

$$m_i = 2^{-\frac{i-1}{r}} m_1 .$$

Proof. We take the partial derivatives of D'_M with respect to m_i, $1 < i < r$. Setting all derivatives to zero, and solving the resulting equations gives the desired results. □

We note that the node sizes that minimize the expected utilization rates are those that minimize the worst-case utilization rates, which will be approximately $2^{-1/r}$.

Corollary 3. For fixed r, the storage utilization rates U_M and U'_M are bounded by

$$U_M \leq \frac{(m_1 + 1)\, G(m_1) - 1}{(m_1 + 1)\, r(2^{1/r} - 1) - 2}$$

$$U'_M \leq \frac{m_1\, G(m_1 - 1)}{m_1\, r(2^{1/r} - 1) - 1} .$$

Proof. Substitution into the result of Theorem 2 with the result of Theorem 3 gives the result. □

Corollary 4. For fixed r, and sufficiently large m_1 and n, the storage utilization rates for B- and B'-trees approach

$$\ln 2/(r(2^{1/r} - 1)).$$

We note that the limit of $r(2^{1/r} - 1)$, as r goes to ∞, is $\ln 2$. This fact is not surprising in view of Corollary 2. For $r = 1, 2, 4, 8, 16$ the asymptotic rates are 69.3%, 84.0%, 91.6%, 95.7%, 97.9%, respectively.

5. Storage utilization of the modification of the Bayer and McCreight overflow technique

We briefly review the definition of B*-trees, from [4]. A B*-tree of order m is a tree satisfying properties 3) - 5) from section 2, plus the following properties:

1*) Every node has at most m sons, except for the root, which may have at most $2\lfloor(2m + 1)/3\rfloor - 1$ sons.
2*) Every node, except for the root and the leaves, has at least $\lfloor(2m + 1)/3\rfloor$ sons.

The worst-case storage utilization is increased from the ratio of approximately 1/2 for B-trees to approximately 2/3 for B*-trees. This is done at a cost of greater overhead for insertion into a node.

We introduce the following modifications to the B*-tree, in order to make an analysis of storage utilization tractable. We group brother nodes together into brother sets of either two or three nodes. An insertion that would cause an overflow in a node is rotated, if possible, to another node in the brother set. If all nodes in a brother set are full, then the records in the brother set are split among a one greater number of nodes. If this split creates four nodes, then these nodes are divided into two brother sets of two nodes each.

In splitting a node, care must be taken not to split a brother set consisting of the node's sons. Hence, the split may not be as even as otherwise possible, in order to preserve brother sets. Thus we introduce a B^+-tree of order m, that satisfies properties 1*), 3) - 5) from section 2, along with

2^+) Every node, except for the root and the leaves, has at least $\lfloor(2m - 2)/3\rfloor$ sons, and is a member of a brother set of either two or three nodes, which is split only in the case of an insertion into a set of three full brother nodes.

Since no information is stored at the leaves (pointed to by null pointers), we can in fact enforce property 2^+) for nodes one level above the (null) leaves. Analogous to the B'-tree, we may define the B^+'-tree. Random B^+-trees and almost random B^+'-trees have definitions parallel to those in section 2.

The analysis for B^+'-trees follows directly from Theorem 2, but we must proceed with the analysis of B^+-trees, in a fashion similar to that of B-trees in section 3. We omit details here, and refer the reader to [7].

Let $U_m^+(n)$ be the ratio of the expected number of records to the expected number of record positions in the lowest level of an n-key random B^+-tree. Let $U_m^{+\prime}(n)$ be the corresponding quantity in an n-key almost random $B^{+\prime}$-tree. Let U_m^+ and $U_m^{+\prime}$ be the corresponding limits.

Theorem 4. Let the asymptotic storage utilization rates U_m^+ and $U_m^{+\prime}$ for B^+- and $B^{+\prime}$-trees, respectively, be defined as above. Then

$$U_m^+ = \frac{1}{m-1}\left[\left(\frac{1}{2m+1} + \frac{1}{3m+1}\right)^{-1} G(3m) - 1\right]$$

$$U_m^{+\prime} = \frac{1}{m-1}\left(\frac{1}{2(m-1)+1} + \frac{1}{3(m-1)+1}\right)^{-1} G(3(m-1)).$$

Proof. $B^{+\prime}$-trees are easy, since we note that a brother set acts as a supernode, taking on two node sizes: $m_1 = 3(m-1)+1$ and $m_2 = 2(m-1)+1$. The number of insertion positions in such a supernode will equal the number of records in it. Hence

$$D_M' = (m-1)\left(\frac{1}{2(m-1)+1} + \frac{1}{3(m-1)+1}\right).$$

Substitution into the result of U_M' from Theorem 2, with $r = 2$, yields the result.

An appeal to Theorem 2 will not work for B^+-trees, since there are either 2 or 3 insertion positions more than the number of records in a supernode. In a fashion analogous to Theorem 1, we get the desired result. ☐

While we were not able to proceed directly from Theorem 2 to Theorem 4 in the case of B^+-trees, the results suggest the (at least) intuitive explanation of a supernode taking on two different node sizes. Hence for both types of trees the results relate in an interesting manner to those of section 4. We conclude this section with the corollary that yields the asymptotic utilization rate of 83.2%.

Corollary 5. Let U_m^+ and $U_m^{+\prime}$ be as in Theorem 4. Then

$$\lim_{m\to\infty} U_m^+ = \lim_{m\to\infty} U_m^{+\prime} = \frac{6}{5}\ln 2.$$

REFERENCES

[1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, The design and analysis of computer algorithms, Addison-Wesley, Reading, Mass. (1974).

[2] R. Bayer and E. McCreight, "Organization and maintenance of large ordered indexes," Acta Informatica 1 (1972) 173-189.

[3] D. E. Knuth, The art of computer programming, Vol. I, Fundamental algorithms, Addison-Wesley, Reading, Mass. (1968).

[4] D. E. Knuth, The art of computer programming, Vol. III, Sorting and searching, Addison-Wesley, Reading, Mass. (1973).

[5] T. Nakamura and T. Mizoguchi, "An analysis of storage utilization factor in block split data structuring scheme," Proceedings of 4th Conference of Very Large Data Bases, Berlin (1978).

[6] A. C. Yao, "On random 2-3 trees," Acta Informatica 9 (1978) 159-170.

[7] G. N. Frederickson, "Improving storage utilization in balanced trees," Department of Computer Science TR CS-79-37, The Pennsylvania State University (1979).

ON REQUIRED SPACE FOR RANDOM SPLIT FILES

TETSUO MIZOGUCHI
Computer Laboratory
Mitsubishi Electric Corporation
325 Kamimachiya, Kamakura city, JAPAN

ABSTRACT

In this paper we analyze the average space required to store the linearly ordered set of data records to a split file, where data records are inserted and deleted randomly. Given block size b (the maximum number of data records in a data block), and deletion/insertion rate r, the asymptotic storage utilization factor (SUF) is approximately,

$$\mathrm{SUF} = \mathrm{SUF}_{r=0} - \frac{r}{(1-r)b} \qquad 0 \leqslant r < 1$$

and for large b,

$$\mathrm{SUF}_{r=0} = \ell n\ 2 = .6931, \quad \mathrm{SUF}_{r=1} = \frac{1}{4\ \ell n\ 2} = .3607.$$

INTRODUCTION

The split file is one of the balanced trees where the height of the tree is balanced dynamically when the insertion and deletion of data records are requested. In order to keep the height of the tree balanced, the split file employs the technique "splitting" of data records in one data block into two data blocks if there is no free space to store the data records of the intended insertion. In the resulted tree the data records at the leaves are in the same level at any time. While the retrieval time delay in the worst case is minimized by splitting, it is considered that the space efficiency could be degraded excessively because of splitting.

In the previous works [4],[5], the storage utilization factor was analyzed when only random insertion is assumed. In this paper we extend the analysis to the case where both random insertion and deletion are assumed with an arbitrary deletion/insertion rate r of $0 \leqslant r \leqslant 1$, although we assume that r stays constant.

In the following sections, we give some preliminaries and the asymptotically expected SUF when the occurrences of random insertion and deletion are equally likely, i.e. r=1, and also when r is less than 1 and constant.

PRELIMINARIES

In this section we define the terms and model to be used in this paper.

Split file A split file consists of index blocks and data blocks. In this paper we assume that there are one or more than one level of index blocks and one level of data blocks. Each entry in an index block holds a pointer to point one of blocks of immediately lower level and a key to identify the range of keys in the pointed blocks. In this paper we analyze the storage utilization of data blocks only, that is , the storage utilization analysis of the lowest level blocks of the split files.

Each data record in a data block is of same physical size in one file, and consists of the ordered pairs (k,d); key and data. In the following we assume that all keys are distinct.

In a split file, data records are always maintained in a linear order defined on the key set of data records. By block size b, we mean that one data block contains at most b data records. If there are m data blocks B_i, $1 \leq i \leq m$ and at most b data records $R_{ij} = (k_{ij}, d_{ij})$ in B_i, $1 \leq j \leq b$, then $k_{ij} < k_{i'j'}$ when $i < i'$ or $i = i'$ and $j < j'$. Suppose data block B_i contains b data records, i.e. no free space in B_i, and an insertion of data record to B_i is requested. Since there is no free space in B_i, the maintenance procedure obtains a free data block B_i' and splits (b+1) data records, say $\lceil (b+1)/2 \rceil$ and $\lfloor (b+1)/2 \rfloor$ each, to two data blocks B_i and B_i', i.e. "splitting". The maintenance procedure returns data block B_i to the free data block set only when all data records in B_i are deleted, that is, the procedure does not perform any tuning operations like merging or distributing data records among neighboring data blocks to improve the space efficiency.

Random insertion and deletion The operation to files is any one of retrieval, modification of data record contents, insertion, or deletion. Since only insertion and deletion among four basic operations are related to the change in space efficiency, in this paper we consider the effect of insertion and deletion, more specifically of random insertion and deletion, on the space efficiency.

By randomness of insertion(deletion), we mean that the probability that an insertion(deletion) request hits a data block is proportional to the number of data records in the data block. This definition differs from the one by Yao[5] and is called "almost-random" in [2]. By event we mean an occurrence of random insertion or random deletion.

By deletion/insertion rate r, we mean

$$r = \frac{\text{the number of random deletions up to event time } t}{\text{the number of random insertions up to event time } t} = \frac{(1-p)t}{pt}$$

where p is the insertion rate over whole events, $1/2 \leq p \leq 1$. Here we assume that r stays constant and $0 \leq r \leq 1$, that is, we assume that the number of data records in a file either increases linearly ($0 \leq r < 1$) or stays constant ($r = 1$). Although the event is a discrete object, we assume that there have been pt random insertions and (1-p)t random deletions up to time t, and at event time (t+1) there will be p random insertion and (1-p) random deletion.

State vector and transition matrix Here we introduce a state vector $A^{(t)}$

$$A^{(t)} = (A_1^{(t)}, A_2^{(t)}, \ldots, A_b^{(t)})^T$$

where $A_i^{(t)}$ is the number of data records containing i data records at time t and b is the block size. By the definition of randomness given above, the transition matrix R is given to determine the state vector $A^{(t+1)}$.

$$A^{(t+1)} = R.A^{(t)} = (I+G+H).A^{(t)}$$

where I is the identity matrix, G is an insertion transition matrix, and H is a deletion transition matrix. The element g_{ij} of G, h_{ij} of H and r_{ij} of R are given as follows.

$$g_{ij} = \begin{cases} -\frac{i}{n}\,p & \text{if } j=i \quad \text{;outgoing when inserted} \\ \frac{i-1}{n}\,p & \text{if } j=i-1 \quad \text{;incoming when inserted} \\ \frac{b}{n}\,p & \text{if } j=b, i=\frac{b}{2},\frac{b}{2}+1 \text{ (b is even)} \quad \text{;incoming when inserted and splitting} \\ \frac{2b}{n}\,p & \text{if } j=b, i=\frac{b+1}{2} \text{ (b is odd)} \quad \text{;incoming when inserted and splitting} \\ 0 & \text{otherwise} \end{cases}$$

$$h_{ij} = \begin{cases} -\frac{i}{n}\,(1-p) & \text{if } j=i \quad \text{;outgoing when deleted} \\ \frac{i+1}{n}\,(1-p) & \text{if } j=i+1 \quad \text{;incoming when deleted} \\ 0 & \text{otherwise} \end{cases}$$

$$R = \left|\begin{array}{ccccccccc} 1-\frac{1}{n} & \frac{2(1-p)}{n} & & & & & & & \\ \frac{p}{n} & 1-\frac{2}{n} & \frac{3(1-p)}{n} & & & & & & \\ & \frac{2p}{n} & \cdot & \cdot & & & & & \\ & & \frac{(\frac{b}{2}-1)p}{n} & 1-\frac{\frac{b}{2}}{n} & \frac{(\frac{b}{2}+1)(1-p)}{n} & & & & \frac{bp}{n} \\ & & & \frac{\frac{b}{2}p}{n} & 1-\frac{\frac{b}{2}+1}{n} & \frac{(\frac{b}{2}+2)(1-p)}{n} & & & \frac{bp}{n} \\ & & & & \cdot & \cdot & \cdot & & \\ & & & & & & \frac{(b-2)p}{n} & 1-\frac{b-1}{n} & \frac{b(1-p)}{n} \\ & & & & & & & \frac{(b-1)p}{n} & 1-\frac{b}{n} \end{array}\right|$$

where n is the number of data records at time t.

In this paper we analyze only the case b=2q for some positive integer q. The case b=2q+1 can be treated similarly.

Storage utilization factor: SUF   The Storage Utilization Factor(SUF) is defined as the ratio of effectively used storage space to the whole available space in a split file, or given a state vector $A^{(t)}$,

$$\mathrm{SUF} = \frac{\sum_{i=1}^{b} i.A_i^{(t)}}{b.\sum_{i=1}^{b} A_i^{(t)}}$$

We are interested in analyzing the asymptotic behavior of SUF for sufficiently large t.

STORAGE UTILIZATION WHEN INSERTION AND DELETION ARE EQUALLY LIKELY

In this section we analyze the asymptotically expected SUF when insertion and deletion are equally likely. Here we first assume that

$$A = \lim_{t \to \infty} A^{(t)}$$

By observing the matrix R given in the previous section, it is clear that A has a solution as follows, when p = 1/2.

$$A_1 = A_2 = . . = A_{\frac{b}{2}}$$

$$\frac{A_{\frac{b}{2}+1}}{A_{\frac{b}{2}}} = \frac{\frac{b}{2}}{\frac{b}{2}+1}$$

$$\frac{A_{b-i}}{A_{b-i-1}} = \frac{i+1}{i+2}\;\frac{b-i-1}{b-i} \qquad 0 \leqslant i \leqslant \frac{b}{2}-2$$

By the definition of SUF,

$$\mathrm{SUF} = \frac{\sum_{i=0}^{\frac{b}{2}-1} \frac{(b-i)(i+1)}{b-i} A_{\frac{b}{2}} + \sum_{j=1}^{\frac{b}{2}} j.A_{\frac{b}{2}}}{b\left[\sum_{i=0}^{\frac{b}{2}-1} \frac{i+1}{b-i} A_{\frac{b}{2}} + \sum_{j=1}^{\frac{b}{2}} A_{\frac{b}{2}}\right]} = \frac{b+2}{4}\;\frac{1}{\frac{b}{2} + \sum_{i=0}^{\frac{b}{2}-1} \frac{i+1}{b-i}}$$

Since for the sufficiently large b,

$$\sum_{i=0}^{\frac{b}{2}-1} \frac{i+1}{b-i} = \left(1-\frac{1}{\frac{b}{2}+1}\right)+\left(1-\frac{3}{\frac{b}{2}+2}\right)+ . . +\left(1-\frac{b-1}{\frac{b}{2}+\frac{b}{2}}\right)$$

$$\simeq \frac{b}{2} - \int_{1}^{\frac{b}{2}} \frac{2x-1}{\frac{b}{2}+x}\,dx = -\frac{b}{2} - 2 + (b+1)\,\ln\left(\frac{b}{\frac{b}{2}+1}\right)$$

SUF for the sufficiently large b is

$$\text{SUF} = \frac{b+2}{4} \frac{1}{-2+(b+1)\ln(\frac{b}{\frac{b}{2}+1})} \simeq \frac{1}{4 \ln 2} = .3607$$

Now we want to show that starting at any initial state A, the expected SUF converges the one given above. Let

$$A' = (1,1, . . ,1, \frac{\frac{b}{2}}{\frac{b}{2}+1}, \frac{\frac{b}{2}-1}{\frac{b}{2}+1}, . . . , \frac{2}{b-1}, \frac{1}{b})^T$$

where the first half elements are unit.

Theorem

For any positive vector A,

$$\lim_{t\to\infty} R^t.A = c'.A'$$

where $c' = \frac{1}{\frac{b}{2}(\frac{b}{2}+1)} \sum_{i}^{b} i.A_i$ and A_i is the i th element of A.

Lemma 1

Let

$$\Delta_i = (0,0,....,0,1,0...,0)^T$$

have a unit at i th position and zeroes elsewhere.

$$\lim_{t\to\infty} R^t.\Delta_i = c''_i.A'$$

where $c''_i = \frac{i}{\frac{b}{2}(\frac{b}{2}+1)}$

Lemma 2

For each i

$$| R.\Delta_i - c''_i.A' | < | \Delta_i - c''_i.A' |$$

Here we omit the proof. Theorem is proved by Lemma 1 and 2 which are also proved easily.

STORAGE UTILIZATION WHEN DELETION IS LESS LIKELY THAN INSERTION

In this section we analyze the asymptotically expected SUF when deletion is less likely than insertion. Here we first assume that A_i of state vector increases linearly as t when t is sufficiently large. Let

$$A_i^{(t)} = a_i.t \qquad 1 \leqslant i \leqslant b$$

Given b and r ($r = (1-p)/p$), we want to get a_i's which satisfy

$$A^{(t+1)} = R.A^{(t)}$$

in order to get SUF as follows.

$$\text{SUF} = \frac{\Sigma\, i.a_i}{b\, \Sigma\, a_i}$$

Since $A^{(t+1)} - A^{(t)} = (G+H).A^{(t)}$,

$$(A^{(t+1)} - A^{(t)})t - (G+H).A^{(t)}.t = A^{(t)} - (G+H).A^{(t)}.t$$

$$= (I - \frac{n}{2p-1}(G+H)).A^{(t)} = 0$$

Let

$$F = \frac{2p-1}{p}(I - \frac{n}{2p-1}(G+H))$$

then

$$F = \begin{vmatrix}
(1+r)+(1-r) & -2r & & & & & & & \\
-1 & 2(1+r)+(1-r) & -3r & & & & & & \\
 & -2 & 3(1+r)+(1-r) & -4r & & & & & \\
 & & \cdot & \cdot & \cdot & & & & \\
 & & & -(\frac{b}{2}-1) & \frac{b}{2}(1+r)+(1-r) & -(\frac{b}{2}+1)r & & & -b \\
 & & & & -\frac{b}{2} & (\frac{b}{2}+1)(1+r)-(1-r) & -(\frac{b}{2}+2)r & & -b \\
 & & & & & & & & \\
 & & & & & & -(b-2) & (b-1)(1+r)+(1-r) & -br \\
 & & & & & & & -(b-1) & b(1+r)+(1-r)
\end{vmatrix}$$

From $F.A^{(t)} = 0$, we get the following, (see APPENDIX for details,)

$$\text{SUF} = \frac{(b+1)(H(b)-H(\frac{b}{2}))}{b - r^{\frac{b}{2}}(\frac{2b}{b+2} + \sum_{i=1}^{\frac{b}{2}} \frac{b(b+1)r^i}{(\frac{b}{2}+i)(\frac{b}{2}+i+1)})} - \frac{r}{(1-r)b}$$

$$\text{SUF}_{r=0} = \frac{b+1}{b}(H(b)-H(\frac{b}{2})) \qquad H(m) = \sum_{i=1}^{m} \frac{1}{i}$$

and for large b and small r, SUF is approximately

$$\text{SUF} = \text{SUF}_{r=0} - \frac{r}{(1-r)b}$$

Here we want to show the linearity of state vector $A^{(t)}$ over time t. Let

$$B^{(t)} = \frac{A^{(t)}}{n} = \frac{A^{(t)}}{(2p-1)t}$$

where n is the number of data records at time t, then

$$B^{(t+1)} = \frac{t}{t+1} R.B^{(t)} = \frac{n}{n+2p-1} R.B^{(t)} = R'.B^{(t)}$$

As in the previous section, we can prove that

$$\lim_{t\to\infty} R'^{t}.B = d.B_0$$

for some constant d, where B_0 satisfies $B_0 = R'.B_0$ and $\sum_{i}^{b} i.B_i^{(t)} = 1$ for any t.

This results show the linearity of $A^{(t)}$ over time t.

CONCLUDING REMARKS

Here we give the resulted SUF of our analysis in the following table.

block size	deletion/insertion rate r = 0	r = 0.1	r = 0.2	r = 0.5	r = 1
b = 4	.7290	.7043	.6879	.5983	.5142
b = 8	.7138	.6999 (≃.6999)	.6828 (≃.6825)	.6009 (≃.5888)	.4377
b =16	.7043	(≃.6974)	(≃.6887)	.6426 (≃.6418)	.4010
b =32	.6989	(≃.6954)	(≃.6911)	(≃.6677)	.3800
b =64	.6961	(≃.6943)	(≃.6922)	(≃.6805)	.3704
b = ∞	.6931	——	——	——	.3607

* the values in the parentheses are the approximated SUFs.

From this results, we can state that when the block size is not small, the SUF degrades only slightly unless the deletion/insertion rate is close to 1.

In this paper we analyze the storage utilization of a file scheme with splitting only(i.e. no overflow,no merging). The analysis of a scheme with overflow and merging under random insertion and deletion is still needed.

REFERENCES

[1] Bayer,R. and E.McCreight
"Organization and maintenance of large ordered indexes"
Acta Informatica, vol.1 (1972) pp173-189

[2] Frederickson,G.N.
"Improving storage utilization in balanced trees" (to appear)

[3] Knuth,D.E.
The art of computer programming,vol.3. sorting and searching
Addison-Wesley,1973
[4] Nakamura,T. and T.Mizoguchi
"An analysis of storage utilization factor in block split data structuring scheme", Proc. of 4th International Conference on Very Large Data Bases, Berlin,1978
[5] Yao,A.C-C.
"On random 2-3 trees"
Acta Informatica, vol.9 (1978) pp159-170

APPENDIX

By definition

$$SUF = \frac{\Sigma\ i.a_i}{b.\ \Sigma\ a_i}$$

and

$$n = (2p-1)t = \frac{1-r}{1+r}\,t = \sum_{i=1}^{b} i.a_i.t$$

then

$$\sum_{i=1}^{b} i.a_i = \frac{1-r}{1+r}$$

Now we need to evaluate $\Sigma\ a_i$.

From $F.A^{(t)} = 0$ (section 4)

$$\begin{array}{lll}
a_1 & -ra_2 & =0 \\
a_2 & -ra_3 & =0 \\
\cdot & & \\
a_{\frac{b}{2}-1} & -ra_{\frac{b}{2}} & =0 \\
a_{\frac{b}{2}} & -ra_{\frac{b}{2}+1} & = \frac{b}{\frac{b}{2}+1}\ a_b \\
a_{\frac{b}{2}+1} & -ra_{\frac{b}{2}+2} & = \frac{b(b+1)}{(\frac{b}{2}+2)(\frac{b}{2}+1)}\ a_b \\
\cdot & & \\
a_{b-2} & -ra_{b-1} & = \frac{b(b+1)}{(b-1)(b-2)}\ a_b \\
a_{b-1} & -ra_b & = \frac{b+1}{b-1}\ a_b
\end{array} \qquad \ldots(1)$$

From equation (1),

$$\sum_{i=1}^{\frac{b}{2}} a_i = \frac{1-r^{\frac{b}{2}}}{1-r}\ a_{\frac{b}{2}} \quad , \quad \sum_{i=\frac{b}{2}}^{b-1} a_i - r \sum_{j=\frac{b}{2}+1}^{b} a_j = a_b(b-1)$$

then

$$\sum_{i=1}^{b} a_i = \frac{b.a_b}{1-r} - \frac{r^{\frac{b}{2}}}{1-r}\ a_{\frac{b}{2}}$$

Let

$$\sum_{i=1}^{b} a_i = k.a_b$$

where

$$k(1-r) = b - r^{\frac{b}{2}} . \frac{a_{\frac{b}{2}}}{a_b}$$

Now it is clear that

$$SUF = \frac{\frac{1-r}{1+r}}{b.k.a_b}$$

Here we introduce a new variable X such that

$$X = b.a_b + \sum_{i=1}^{b-1} i(a_i - r.a_{i+1})$$

From equation (1),

$$X = b(b+1)\left(\frac{1}{b} + \frac{1}{b-1} +. \; . + \frac{1}{\frac{b}{2}+1}\right) a_b$$

$$= b(b+1)a_b \; (H(b)-H(\tfrac{b}{2}))$$

on the other hand,

$$X = \sum_{i=1}^{b} i.a_i - r \sum_{i=2}^{b} i.a_i + r \sum_{i=2}^{b} a_i$$

$$= \frac{(1-r)^2}{1+r} + r \sum_{i=1}^{b} a_i$$

Here we get

$$a_b = \frac{\frac{(1-r)^2}{1+r}}{b(b+1)(H(b)-H(\frac{b}{2}))-kr}$$

substituting a_b,

$$SUF = \frac{(b+1)(H(b)-H(\frac{b}{2}))}{k(1-r)} - \frac{r}{(1-r)b}$$

k(1-r) is given by computing $a_{\frac{b}{2}} / a_b$ from equation (1),

$$a_{\frac{b}{2}}/a_b = \frac{2b}{b+2} + b(b+1) \sum_{i=1}^{\frac{b}{2}} \frac{r^i}{(\frac{b}{2}+i)(\frac{b}{2}+i+1)}$$

AN EVALUATION OF THE WYNER-ZIV RATE REGION

K.M. MACKENTHUN, JR.
College of Engineering
University of South Carolina
Columbia, SC 29208

ABSTRACT

Wyner and Ziv considered the problem of source encoding and decoding when the decoder knows a random variable Y correlated with the source X. The least rate R for which the system can operate with fidelity d is given in terms of an information theoretic minimization R*(d). In this correspondence, explicit upper and lower bounds to R*(d) are given for an important class of correlated sources. The upper and lower bounds have been evaluated for several members of the class and shown to agree in these cases.

I. INTRODUCTION

Wyner and Ziv considered the problem of source encoding with a fidelity criterion when the decoder has access to side information about the source [1], as shown in Figure 1. The sequence $\{(X_k,Y_k)\}$ represents independent copies of a pair of dependent random variables (X,Y) which take values in the finite sets $\mathcal{X},\mathcal{Y}$, respectively. The encoder output is a binary sequence which appears at a rate R bits per input symbol. The decoder output is a sequence $\{\hat{X}_k\}$ which takes values in the finite reproduction alphabet $\hat{\mathcal{X}}$. The encoding and decoding is done in blocks of length n, and the fidelity cirterion is the expectation of

$$\frac{1}{n}\sum_{k=1}^{n} D(X_k,\hat{X}_k)$$

where $D(x,\hat{x})\geq 0$, $x\varepsilon\mathcal{X}$, $\hat{x}\varepsilon\hat{\mathcal{X}}$, is a given distortion function. R*(d) is the minimum rate for which the system of Figure 1 can operate with average distortion $E[\Sigma D(X_k,\hat{X}_k)]/n$ arbitrarily close to d when n is sufficiently large. In this correspondence, all logarithms are taken to the base 2 and $|A|$ is the cardinality of the set A.

Wyner and Ziv showed that for $d\geq 0$,

$$R^*(d)=\inf[I(X;Z) - I(Y;Z)]$$

where the inf is over all random variables Z such that

a) Y - X - Z is a Markov chain,

b) Z takes values in $\mathcal{Z}$ where $|\mathcal{Z}|\leq|\mathcal{X}| + 1$,

c) there exists a function f: $\mathcal{Y}\times\mathcal{Z} \rightarrow \hat{\mathcal{X}}$ such that $ED(X,\hat{X})\leq d$ where $\hat{X}=f(Y,Z)$.

Because the preceding representation of R*(d) is quite abstract, it is important to give explicit evaluations of R*(d) for particular sources. In [1], Wyner and Ziv evaluated R*(d) for the correlated source (X,Y) where X is the unbiased input to a binary symmetric channel and Y is the output; the

---

The author is with the College of Engineering, University of South Carolina, Columbia, SC 29208. This work was supported in part by NSF Grant ENG79-08073.

Hamming distortion measure is used. In this correspondince, the source (X,Y) shown in Figure 2 is considered: X is uniformly distributed on an alphabet of M symbols; Y=X with probability $1-P_e$ where $P_e \le 1/2$; and, if $Y \ne X$, Y is equally likely among the other M-1 symbols (note that for M=2, this is the Wyner-Ziv example). This is an important case to consider in side information theory and also has application in deriving performance bounds for error detecting codes [2]. In this correspondence, explicit upper and lower bounds to R*(d) are given for source (X,Y) for the Hamming distortion measure

$$d_H(x,\hat{x}) = \begin{cases} 0, & x=\hat{x} \\ 1, & x\ne\hat{x}, \end{cases}$$

and distortion d satisfying $0 \le d \le P_e$. It is clear that $R^*(d)=0$ for $d \ge P_e$. Calculations of the upper and lower bounds are made for several values of P_e and M; in all cases the upper and lower bounds agree.

II. LOWER BOUND

Step 1. Consider a set $\mathcal{Z}$ having elements $z_1, \ldots z_{M+1}$ and, for d satisfying $0 \le d \le P_e$, the function

$$\tilde{R}(d) = \inf \sum_{i=1}^{M+1} \lambda_i [H(Y|Z=z_i) - H(X|Z=z_i)]$$

where the inf is over all conditional distributions $p(x|z_i)$, i=1, ... M+1 and nonnegative numbers λ_i, i=1, ... M+1 such that

$$\text{a}')\ \sum_{i=1}^{M+1} \lambda_i = 1,$$

b') there exist functions $f_i(\cdot,z_i): \mathcal{Y}\times\mathcal{Z}\to\hat{\mathcal{X}}$, i=1, ... M+1 such that

$$\sum_{i=1}^{M+1} \lambda_i E d_H(X, f_i(Y,z_i)) \le d.$$

Claim 1. $\tilde{R}(d) \le R^*(d)$.

Proof. For any Z, f satisfying (a), (b), (c), we associate the natural conditional distribution $p(x|z_i)$, i=1, ... M+1; positive numbers $\lambda_i = p(z_i)$, i=1, ... M+1; and functions $f_i(\cdot,z_i) = f(\cdot,z_i)$, i=1, ... M+1. When $p(z_i)=0$, we take arbitrary choices for $p(x|z_i)$ and $f_i(\cdot,z_i)$. We see that (a'), (b') are satisfied and

$$I(X;Z) - I(Y;Z) = \sum_{i=1}^{M+1} \lambda_i [H(Y|Z=z_i) - H(X|Z=z_i)]$$

so that $\tilde{R}(d) \le R^*(d)$.

Step 2. We want to find $\tilde{R}(d)$ so that we may underbound R*(d). As a first step consider a particular $z \in \mathcal{Z}$ and, for d satisfying $0 \le d \le P_e$, the function

$$R_z(d) = \inf [H(Y|Z=z) - H(X|Z=z)]$$

where the inf is over all conditional distributions $\alpha_i = p(x_i|z)$, i=1,...M such that there exists a function $f: \mathcal{Y} \times z \to \hat{X}$ satisfying

$$Ed_H(X,f(Y,z)) \le d.$$

The function $R_z(d)$ does not depend on which particular $z \in Z$ is chosen so that we have the following equivalent formulation of $\tilde{R}(d)$

$$\tilde{R}(d) = \inf \sum_{i=1}^{M+1} \lambda_i R_z(d_i)$$

where the inf is over all nonnegative numbers λ_i, d_i, i=1, ... M+1 such that

$$\text{a'')} \quad \sum_{i=1}^{M+1} \lambda_i = 1,$$

$$\text{b'')} \quad \sum_{i=1}^{M+1} \lambda_i d_i \le d.$$

Step 3. We now prove two useful results concerning $R_z(d)$. It will be convenient to define the function $g: [0,\infty) \to (-\infty,\infty)$ by

$$g(\alpha) = -\left[\frac{P_e}{M-1} + \left(1-\frac{MP_e}{M-1}\right)\alpha\right] \log\left[\frac{P_e}{M-1} + \left(1-\frac{MP_e}{M-1}\right)\alpha\right] + \alpha\log\alpha.$$

Claim 2.

$$H(Y|Z=z) - H(X|Z=z) = \sum_{i=1}^{M} g(\alpha_i).$$

Claim 3. For $0 \le \alpha \le 1$, $g(\alpha)$ is a convex function of α. Claim 2 is proven by calculating $H(Y) - H(X)$ when the distribution of X is given by $\alpha_1, \alpha_2, \ldots \alpha_M$. Claim 3 is proven by showing that the second derivative of $g(\alpha)$ is nonnegative.

Step 4. We now study $R_z(d)$. Using Claim 2, note that we may rewrite $R_z(d)$ as

$$R_z(d) = \inf \sum_{i=1}^{M} g(\alpha_i)$$

where the inf is over all conditional distributions $\alpha_i = p(x_i|z)$, i=1, ... M such that there exists a function $f: \mathcal{Y} \times z \to \hat{X}$ satisfying

$$E\, d_H(X,f(Y,z)) \le d.$$

Because of the symmetry of the correlated source, it is sufficient to minimize $\Sigma\, g(\alpha_i)$ over the set A of vectors $\underline{\alpha} = (\alpha_1, \alpha_2, \ldots \alpha_M)$ where $\alpha_1 \ge \alpha_2 \ge \ldots \alpha_M$. Divide the set A into M equivalence classes, A_j, j=1,...M by the following relations:

$$(1-P_e)\,\alpha_i \ge \frac{P_e}{M-1}\,\alpha_1 \qquad ,i=1, \ldots j \qquad (1)$$

$$(1-P_e)\ \alpha_i < \frac{P_e}{M-1}\ \alpha_1 \qquad ,i=j+1,\ldots M \tag{2}$$

Consider $R_z^j(d)$, the minimization of $R_z(d)$ restricted to equivalence class A_j. It is clear that for any conditional distribution $\underline{\alpha} \varepsilon A_j$, the $f_{\underline{\alpha}}$ which minimizes $E\, d_H(X,f_{\underline{\alpha}}(Y,z))$, the Bayes cost, is obtained by letting $f_{\underline{\alpha}}(y_i,z)$ be an x_k satisfying

$$p(x_k \mid y_i,z) \geq p(x_j \mid y_i,z) \qquad ,j=1,\ldots M$$

or equivalently

$$p(y_i \mid x_k)\ \alpha_k \geq p(y_i \mid x_j)\ \alpha_j \qquad ,j=1,\ldots M$$

Because $\alpha_1 \geq \alpha_2 \geq \ldots \alpha_M$, we may rewrite our Bayes rule as

$$f_{\underline{\alpha}}\ (y_i,z) = \begin{cases} x_i & \text{if } (1-P_e)\ \alpha_i \geq \dfrac{P_e}{M-1}\ \alpha_1 \\ x_1 & \text{if } (1-P_e)\ \alpha_i < \dfrac{P_e}{M-1}\ \alpha_1 \end{cases}$$

which takes the following form for $\underline{\alpha}\varepsilon A_j$

$$f_{\underline{\alpha}}\ (y_i,z) = \begin{cases} x_i & , i=1,\ \ldots\ j \\ x_1 & , i=j+1,\ldots M \end{cases} \tag{3}$$

Thus, to find $R_z^j(d)$ it is sufficient to minimize $\Sigma\ g\ (\alpha_i)$ over all conditional distributions $\underline{\alpha} = (\alpha_1,\ldots\alpha_M)\ \varepsilon A_j$ such that

$$E\ d_H(X,f_{\underline{\alpha}}(Y,z)) \leq d \tag{4}$$

where $f_{\underline{\alpha}}$ satisfies (3). If there is no $\underline{\alpha}\varepsilon A_j$ such that $Ed_H(X,f_{\underline{\alpha}}(Y,z)) \leq d$, we define $R_z^j(d) = +\infty$.

Note that $R_z^j(d)$ is decreasing. In the region where $R_z^j(d)$ is finite, Claim 3 shows that $R_z^j(d)$ is strictly decreasing so that we may replace the inequality in (4) by equality. This gives the following.

Claim 4. $R_z^j(d) < +\infty$ if and only if there is an $\underline{\alpha}\ \varepsilon\ A_j$ satisfying

$$E\ d_H(X,f_{\underline{\alpha}}(Y,z)) = d \tag{5}$$

where $f_{\underline{\alpha}}$ is given in (3). If $R_z^j(d) < +\infty$, then

$$R_z^j(d) = \inf\ \Sigma\ g(\alpha_i)$$

where the inf is over all conditional distributions $\underline{\alpha} = (\alpha_1, \ldots \alpha_M) \varepsilon A_j$ satisfying (5).

We know evaluate $R_z^1(d)$. For $\underline{\alpha} \varepsilon A_1$,

$$Ed_H(X, f_{\underline{\alpha}}(Y,z)) = \sum_{i=1}^{M} E[d_H(X, f(Y,z)) \mid X = x_i] P\{X = x_i\}$$

$$= \alpha_2 + \alpha_3 = \ldots \alpha_M \tag{6}$$

From Claim 4 and (6), we see that $R_z^1(d) < +\infty$ for $0 \leq d \leq P_e$. Moreover, (6), Claim 4, and the convexity of $\Sigma\, g(\alpha_i)$ imply that the unique $\underline{\alpha} \varepsilon A_1$ which minimizes $R_z^1(d)$ is given by

$$\alpha_2 = \alpha_3 = \ldots \alpha_M = \frac{d}{M-1} \tag{7a}$$

$$\alpha_1 = 1 - d. \tag{7b}$$

Therefore for $0 \leq d \leq P_e$,

$$R_z^1(d) = g(1-d) + (M-1)\, g\left(\frac{d}{M-1}\right) \tag{8}$$

We now find $R_z^M(d)$. For $\underline{\alpha} \varepsilon A_M$,

$$Ed_H(X, f_{\underline{\alpha}}(Y,z)) = P_e(\alpha_1 + \alpha_2 + \ldots \alpha_M) = P_e \tag{9}$$

From Claim 4 and (9), we see that $R_z^M(d) = +\infty$ for $0 \leq d < P_e$. Equation (9) and the convexity of $\Sigma\, g(\alpha_i)$ imply that the unique $\underline{\alpha} \varepsilon A_M$ which minimizes $R_z^M(P_e)$ is given by

$$\alpha_1 = \alpha_2 = \ldots \alpha_M = \frac{1}{M} \tag{10}$$

Therefore,

$$R_z^M(P_e) = Mg\left(\frac{1}{M}\right) = 0. \tag{11}$$

We now seek a representation for $R_z^j(d)$ for j satisfying $1 < j < M$. For $\underline{\alpha} \varepsilon A_j$,

$$E\, d_H(X, f_{\underline{\alpha}}(Y,z)) = \frac{(j-1)P_e}{M-1}\, \alpha_1 + P_e(\alpha_2 + \ldots \alpha_j) + (\alpha_{j+1} + \ldots \alpha_M) \tag{12}$$

Equation (12), Claim 4, and the convexity of $\Sigma\, g(\alpha_i)$ imply that to find $R_z^j(d)$ we need consider only $\underline{\alpha} \varepsilon A_j$ satisfying

$$\alpha_2 = \ldots = \alpha_j \tag{13}$$

$$\alpha_{j+1} = \ldots = \alpha_M \tag{14}$$

$$\frac{(j-1)P_e}{M-1}\alpha_1 + (j-1)P_e\alpha_L + (M-j)\alpha_s = d. \tag{15}$$

In (15), we have used $\alpha_L = \alpha_2 = \ldots\alpha_j$ and $\alpha_s = \alpha_{j+1} = \ldots\alpha_M$. For vectors $\underline{\alpha}\varepsilon A_j$ satisfying (13) and (14), we have that

$$\alpha_1 + (j-1)\alpha_L + (M-j)\alpha_s = 1 \tag{16}$$

$$(1-P_e)\alpha_L \geq \frac{P_e}{M-1}\alpha_1 \tag{17}$$

$$(1-P_e)\alpha_s < \frac{P_e}{M-1}\alpha_1. \tag{18}$$

Thus $R_z^j(d)$ is found by minimizing

$$g(\alpha_1) + (j-1)g(\alpha_L) + (M-j)g(\alpha_s) \tag{19}$$

over all α_1, α_L, α_s satisfying (15)-(18).

This minimization can be somewhat simplified as follows; note that using (16), (19) can be rewritten as

$$g(1-(j-1)\alpha_L-(M-j)\alpha_s) + (j-1)g(\alpha_L) + (M-j)g(\alpha_s) \tag{20}$$

and (15) becomes

$$\frac{(j-1)P_e}{M-1} + \left[1 - \frac{j-1}{M-1}\right](j-1)P_e\alpha_L + \left[1 - \frac{(j-1)P_e}{M-1}\right](M-j)\alpha_s = d. \tag{21}$$

In addition, (16) gives

$$j\alpha_L + (M-j)\alpha_s \leq 1 \tag{22}$$

since $\alpha_1 \geq \alpha_L$. Substituting (16) into (17) and (18) gives the following 2 inequalities in α_s, α_L:

$$(1-P_e)\alpha_L \geq \frac{P_e}{M-1}(1-(j-1)\alpha_L - (M-j)\alpha_s) \tag{23}$$

$$(1-P_e)\alpha_s < \frac{P_e}{M-1}(1-(j-1)\alpha_L - (M-j)\alpha_s). \tag{24}$$

Therefore for j satisfying $1<j<M$, $R_z^j(d)$ is found by minimizing (20) over the set $\underset{\sim}{d}_j$ of (α_L, α_s) satisfying (21) and inequalities (22), (23), and (24).

The set $\underset{\sim}{d}_j$ is illustrated in Figure 3. Inequalities (22), (23), and

(24) define half-planes in the (α_L,α_s) plane. The 3 half-planes intersect to give the shaded region shown in Figure 3; the 3 solid oblique lines which bound the shaded region are the half-plane boundaries defined by (22), (23), and (24). The equation of the dashed line is given in (21). The set $\underset{\sim}{d}_j$ then corresponds to the dashed line segment lying within the shaded region. $\underset{\sim}{d}_j$ is a closed convex region of the (α_L,α_s) plane and by Claim 3, (20) is a convex function of (α_L,α_s). Therefore the infimum is attained by a unique $(\alpha_L{*}, \alpha_s{*}) \varepsilon \underset{\sim}{d}_j$. We know that (α_L^*,α_s^*) lies on the boundary of $\underset{\sim}{d}_j$ or lies in the interior of $\underset{\sim}{d}_j$ and is the unique solution to the equation

$$G(\alpha_L,\alpha_s) = 0$$

where $G(\alpha_L,\alpha_s)$ is the directional derivative of (20) in the direction defined by line segment $\underset{\sim}{d}_j$. In practice, the solution to (25) or the equivalent Lagrange multiplier formulation is difficult to obtain becuase of the complexity of $g(\alpha)$. However, the minimum of a convex function over a closed convex set can be readily found by computer programming techniques. Lastly, note from Figure 3 that $R_z^j(d) < +\infty$ if and only if d satisfies

$$d \geq \frac{(j-1)P_e}{M-1} + \left[1 - \frac{j-1}{M-1}\right] \frac{(j-1)P_e^{\,2}}{(M-1) - (M-j)P_e} \qquad (26)$$

Summarizing our results, we have

Claim 5. $R_z(d) = \min\ \{ R_z^j(d) \mid 1 \leq j \leq M \}$

where $R_z^1(d)$ is given in (8), $R_z^M(d)$ is given in (11) and $R_z^j(d)$ for $1<j<M$ is found by minimizing (20) over the set $\underset{\sim}{d}_j$.

Step 5. Using Claim 5 in the definition of R(d) given in Step 2 results in a standard variational problem and gives the following lower bound.

Lower bound. $\tilde{R}(d) = \underline{co}\ (R_z^1, R_z^2, \ldots R_z^M)\ (d)$ where $\underline{co}\ (R_z^1, R_z^2, \ldots R_z^M)\ (d)$ is the lower convex hull of $R_z^1(d), R_z^2(d), \ldots R_z^M(d)$.

III. UPPER BOUND

To derive the upper bound, we evaluate I(X;Z) - I(Y;Z) for two different Z,f pairs staisfying (a), (b), and (c).

Step 1. Let Z be connected to X by the channel shown in Figure 2 with parameter d instead of P_e. Then Y - X - Z is a Markov chain and by (7) $I(X;Z) - I(Y;Z) = R_z^1(d)$. Let $\hat{X} = f(Y,Z) = Z$. Then $E\, d_H(X,\hat{X}) = d$ so that

$$R^*(d) \leq I(X;Z) - I(Y;Z) = R_z^1(d).$$

Step 2. Let Z be degenerate, i.e., $|Z| = 1$, and let $\hat{X}=f(Y,Z)=Y$. Then $E\, d_H(X,\hat{X}) = P_e$ so that

$$R^*(P_e) \leq I(X;Z) - I(Y;Z) = 0 = R_z^M(P_e).$$

Step 3. Since $R^*(d)$ is convex [1], we have

Upper Bound. $R^*(d) \leq \underline{co}\ (R_z^1, R_z^M)\ (d)$.

IV. EVALUATION OF THE UPPER AND LOWER BOUND

The importance of the upper and lower bound lies in the fact that they can be shown to agree for a large range of P_e and M. $R_z^2(d), \ldots R_z^{M-1}(d)$ were evaluated for $M = 3,4,5,\ldots 10$ and $M=100$ with $P_e = .005, .05$, and .5 using a computer program to find the minimum of the convex function in (20) over the convex set $\underset{\sim}{d}_j$. For these values of M and P_e, it was found that $\underline{co}\ (R_z^1, R_z^2, \ldots R_z^M)\ (d) = \underline{co}(R_z^1, R_z^M)\ (d)$ so that the upper and lower bounds agree and $R^*(d) = \underline{co}\ (R_z^1, R_z^M)\ (d)$. As an example, $R_z^1(d), R_z^2(d), \ldots R_z^5(d)$ are shown in Figure 4 for M=5 and $P_e=.05$. The dashed line is $\underline{co}\ (R_z^1, R_z^5)\ (d)$ which can be seen to equal $\underline{co}\ (R_z^1, R_z^2, \ldots R_z^5)\ (d)$. We conjecture that $R^*(d) = \underline{co}\ (R_z^1, R_z^M)\ (d)$ for any $M \geq 2$ and P_e satisfying $0 < P_e \leq 1/2$.

When $R^*(d) = \underline{co}\ (R_z^1, R_z^M)\ (d)$, the optimum coding-decoding procedure for low distortion is to have the decoder ignore side information and use only source information; for distortion P_e to have the decoder ignore source information and use only side information; and for intermediate values of distortion to time share the two schemes. It is an interesting open problem to characterize the class of correlated sources for which the optimum performance is obtained by time sharing strictly source information with strictly side information. In general, there exist correlated sources for which the optimum coding-decoding scheme is not a time-sharing scheme; for example, this is true when the source has $p(x|y) = 1$ for some x,y.

REFERENCES

[1] A. D. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," IEEE Trans. Inform. Theory, Vol. IT-22, pp. 1-10, January 1976.

[2] K. M. Mackenthun Jr., "Noiseless coding of error detection information," submitted to IEEE Trans. Inform. Theory.

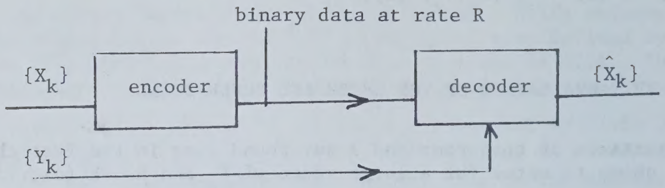

binary data at rate R
{X_k}
encoder
decoder
{X̂_k}
{Y_k}

Figure 1. Source coding with side information at the decoder.

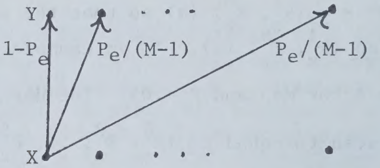

Y
1-P_e
P_e/(M-1)
P_e/(M-1)
X

Figure 2. X is uniformly distributed on M symbols; Y=X with probability $1-P_e$; if $Y \neq X$, Y is equally likely among the other M-1 symbols.

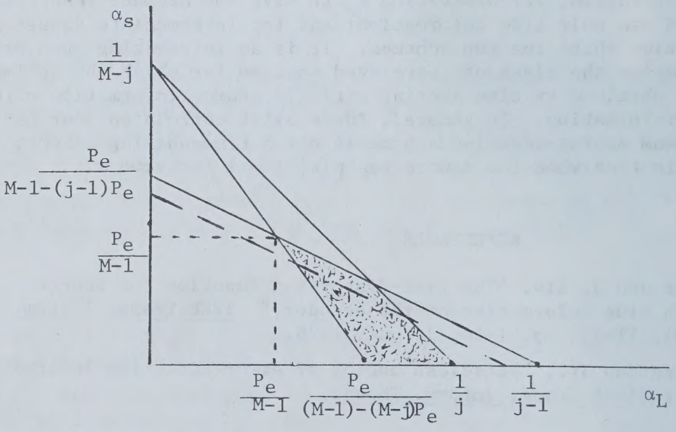

α_s
1/(M-j)
P_e/(M-1-(j-1)P_e)
P_e/(M-1)
P_e/(M-1)
P_e/((M-1)-(M-j)P_e)
1/j
1/(j-1)
α_L

Figure 3. The region of the (α_L, α_s) plane showing the set $\underset{\sim}{d}_j$, the portion of the dashed line lying within the dotted region.

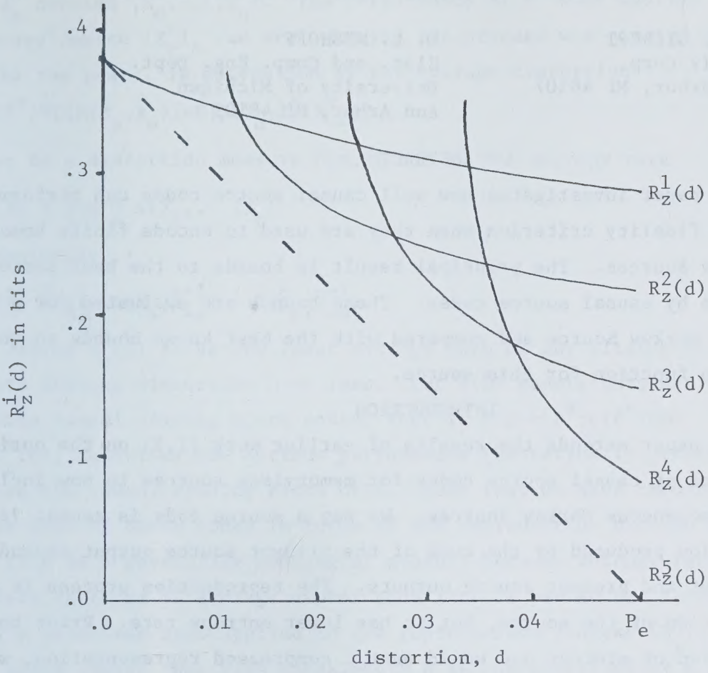


Figure 4. Evaluation of the lower bound for M=5 and P_e=.05. Dashed line is co (R_Z^1, R_Z^5) (d).

BOUNDS TO THE PERFORMANCE OF CAUSAL CODES FOR MARKOV SOURCES

R. K. GILBERT
Bendix Corp.
Ann Arbor, MI 48107

D. L. NEUHOFF
Elec. and Comp. Eng. Dept.
University of Michigan
Ann Arbor, MI 48109

ABSTRACT

This paper investigates how well causal source codes can perform relative to a fidelity criterion when they are used to encode finite homogeneous Markov Sources. The principal result is bounds to the best performance achievable by causal source codes. These bounds are evaluated for a binary symmetric Markov Source and compared with the best known bounds to the rate distortion function for this source.

INTRODUCTION

This paper extends the results of earlier work [1,2] on the optimal performance of causal source codes for memoryless sources to now include finite homogeneous Markov Sources. We say a source code is causal if the reproduction produced by the code of the present source output depends only on the past and present source outputs. The reproduction process is an approximation of the source, but it has lower entropy rate. Prior to transmission or storage one would want a compressed representation, which by the Noiseless Coding Theorem can be performed by an invertible mapping. We are not requiring this mapping to be causal.

In this paper we present bounds on the optimum performance theoretically attainable by causal source codes for finite homogeneous Markov Sources. These bounds are evaluated for a binary symmetric Markov Source and compared with the best known bounds to the rate distortion function R(D), for this source. For the particular example presented, significant performance improvements can be obtained by using optimal noncausal codes.

We shall define a Markov Source to a be a finite, irreducible, aperiodic Markov chain with stationary distribution. Such processes are not only ergodic, but are also mixing [3, pp. 30-33]. Specifically, let $\{X_k\}=\{X_k;k=-\infty,\ldots,\infty\}$ be a sequence of random variables whose ranges are contained in a finite set $A=\{a_1,\ldots,a_J\}$ with stationary transition probabilities,

$$\Pr\{X_{n+1}=a_j|X_n=a_i,X_{n-1}=a_{i_{n-1}},\ldots,X_0=a_{i_o}\}=\Pr\{X_{n+1}=a_j|X_n=a_i\}$$

for every n and every sequence $(a_{i_{n-1}},\ldots,a_{i_o})$ in A^n.

A stationary (or sliding block) causal code is characterized by a function $f^s:A^o_{-\infty}\to A$. The reproduction of the source output X_k is $\hat{X}_k=f^s(X^k_{-\infty})$, where X_m^n denotes $(X_m,\dots,X_n)$. The performance of f^s when applied to the stationary source $\{X_k\}$, (we are assuming the process was started infinitely far into the past), is determined by the average distortion

$$\rho(f^s)\triangleq E[\rho(X_o,\hat{X}_o)]=E[\rho(X_o,f^s(X_{-\infty}^o)]$$

relative to a distortion measure $\rho(x,\hat{x})$ and by the entropy rate

$$H(\hat{X}) \triangleq \lim_{n\to\infty} \frac{1}{n} H(\hat{X}_1,\dots,\hat{X}_n) \qquad (1)$$

or equivalently

$$H(\hat{X}) \triangleq \lim_{n\to\infty} H(\hat{X}_o|\hat{X}_{-n}^{-1}) = H(\hat{X}_o|\hat{X}_{-\infty}^{-1}).$$

We now define $r_c^s(D)$ to be the least entropy rate of any sliding block causal code with average distortion D or less. Let $\Psi(D)$ denote this set of D-admissible causal sliding block codes, that is $\Psi(D)=\{f^s:\rho(f^s)\le D\}$. The function, $r_c^s(D)$, describes the optimum performance theoretically attainable and is called the causal sliding block OPTA. Note that we have chosen to measure the rate of these codes in terms of the reproduction process entropy rate. This is a physically meaningful measure because entropy rate is the least rate in terms of average number code letters per source letter attainable by a noiseless code applied to the reproduction process $\{\hat{X}_k\}$.

A block causal code with blocklength N is characterized by a set of N functions $f_n^b:A_1^n\to A$, $n=1,2,\dots,N$. The reproduction of X_{kN+j} produced by the code is

$$\hat{X}_{kN+j}=f_j^b(X_{kN+1}^{kN+j}), \text{ where } k=-\infty,\dots,\infty \text{ and } j=1,\dots,N.$$

We let $f^b(x_1^N)\triangleq(f_1^b(x_1),\dots,f_N^b(x_1^N))$, and use f^b to denote the code. The performance of f^b is determined by the average distortion

$$\rho(f^b)\triangleq E[\frac{1}{N}\sum_{i=1}^{N}\rho(X_i,\hat{X}_i)] = \frac{1}{N}\sum_{i=1}^{N}E\rho(X_i,f_i^b(X_1^i))$$

and the entropy rate given by (1). Now let $r_c^b(D)$ denote the least rate of any block causal code with average distortion D or less.

The following result relates the OPTAs for these two classes of codes.

Theorem: Let $\{X_n\}$ be a finite, mixing process, then $r_c^b(D)=r_c^s(D)$.

Because our primary objective is to present numerical bounds to $r_c^s(D)$ and the proof of this theorem is rather involved, we will not include the proof in this paper. The proof of $r_c^b(D)\ge r_c^s(D)$ uses an argument similar to one used in [2] which uses causal block codes to create causal sliding block codes with nearly identical performance. The proof of $r_c^b(D)\le r_c^s(D)$ uses an argument similar to one used in [3] which uses a causal sliding block code to create a causal block code with very nearly identical performance.

We would like to be able to evaluate either $r_c^b(D)$ or $r_c^s(D)$ directly. In this paper we compute upper and lower bounds to $r_c^b(D)$ and $r_c^s(D)$ for Markov Sources. Previous work [1,2] has evaluated these OPTA for stationary, memoryless sources.

UPPER BOUND TO $r_c^b(D)$

We can upper bound $r_c^b(D)$ by considering specific causal block codes. Let us consider a class of causal block codes of blocklength N. For any integer $k\varepsilon(0,1,\ldots,N-1)$ let $\underset{\sim}{f}^b(N,k)=(\underset{\sim}{f}_1^b,\ldots,\underset{\sim}{f}_N^b)$ where

$$\underset{\sim}{f}_i^b(x_1^i) = \begin{cases} x_i & i=1,\ldots,N-k \\ x_{N-k} & i=N-k+1,\ldots,N. \end{cases}$$

These codes simply replace the last k letters in each N-block with the (N-k)th letter of the block. The performance of $\underset{\sim}{f}^b(N,k)$ codes can be easily evaluated for Markov Sources because the output process $\{\hat{X}_n\}$ also has the Markov property, i.e.,

$$\Pr\{\hat{X}_j=\hat{x}_j|\hat{X}_{-\infty}^{j-1}=\hat{x}_{-\infty}^{j-1}\}=\Pr\{\hat{X}_j=\hat{x}_j|\hat{X}_{j-1}=\hat{x}_{j-1}\}. \tag{2}$$

Note that the transition probabilities in (2) are non-homogeneous. Nevertheless it is sufficient for showing that the performance of $\underset{\sim}{f}^b(N,k)$ is given by

$$\rho(\underset{\sim}{f}^b(N,k)) = \frac{1}{N}\sum_{i=N-k+1}^{N} E\rho(X_i,X_{N-k}) \tag{3}$$

where $\rho(x,x)=0$ and

$$r(\underset{\sim}{f}^b(N,k)) = \frac{N-k+1}{N}H(X_2|X_1) + \frac{1}{N}H(X_{k+2}|X_1). \tag{4}$$

The $\underset{\sim}{f}^b(N,k)$ codes provide upper bounds to $r_c^b(D)$ at a countable number of points D from 0 to D_{max}, where $D_{max}=\inf_{a\varepsilon A} E\rho(X_o,a)$. The latter performance results from the trivial code $\hat{X}_n=a*$ for all n, where a* achieves the infimum. By time sharing these codes we can obtain an upper bound to $r_c^b(D)$ for any distortion between 0 and D_{max}.

In particular, let us consider time sharing $\underset{\sim}{f}_1^b(N_1,k_1)$ and $\underset{\sim}{f}_2^b(N_2,k_2)$. Let $\hat{f}^b=(\hat{f}_1^b,\ldots,\hat{f}_{N_1+N_2}^b)$ be a causal block code of length N_1+N_2 that time shares $\underset{\sim}{f}_1^b(N_1,k_1)$ and $\underset{\sim}{f}_2^b(N_2,k_2)$. In particular

$$\hat{f}_i^b(x_1^i)\overset{\Delta}{=}\begin{cases} \underset{\sim}{f}_{1,i}^b(x_1^i), & i=1,\ldots,N_1 \\ \underset{\sim}{f}_{2,i-N_1}^b(x_{N_1+1}^i), & i=N_1+1,\ldots,N_1+N_2. \end{cases}$$

The performance of $\hat{f}_b$ applied to Markov Sources can be shown to be

$$\rho(\hat{f}_b)=\frac{N_1}{N_1+N_2}\rho(\underset{\sim}{f}_1^b(N_1,k_1)) + \frac{N_2}{N_1+N_2}\rho(\underset{\sim}{f}_2^b(N_2,k_2)) \tag{5}$$

and

$$r(\hat{f}_b)=\frac{N_1}{N_1+N_2}\, r(\underset{\sim}{f}{}_1^b(N_1,k_1)) + \frac{N_2}{N_1+N_2}\, r(\underset{\sim}{f}{}_2^b(N_1,k_1)) \tag{6}$$

which is a point on the straight line joining $[\rho(\underset{\sim}{f}{}_1^b),r(\underset{\sim}{f}{}_1^b)]$ and $[\rho(\underset{\sim}{f}{}_2^b),r(\underset{\sim}{f}{}_2^b)]$. This evaluation is possible because property (2) also holds for these time shared codes.

Equations (3)-(6) were evaluated for a binary symmetric Markov Source Hamming distortion and $\underset{\sim}{f}{}^b(N,k)$ codes with the following values: (a) k=0 (all N give the same code); and (b) N-k=1 and N=2,3,4... . It was not necessary to evaluate $\underset{\sim}{f}{}^b(N,k)$ codes for N-k≥2, because each such code is a time sharing of the code $f^b(N-k-1,0)$ with $\underset{\sim}{f}{}^b(k+1,k)$. See Figure 1.

LOWER BOUND TO $r_c^s(D)$

Lemma: For a finite alphabet stationary process $\{X_n\}$ and a process $\{\hat{X}_n\}$ derived from $\{X_n\}$ by f^s, a causal sliding block code,

$$H(\hat{X}_o|\hat{X}_{-\infty}^{-1})\ge H(\hat{X}_o|\hat{X}_{-n+1}^{-1},X_{-\infty}^{-n})\ge \frac{1}{n}\, H(\hat{X}_{-n+1}^{o}|X_{-\infty}^{-n}). \tag{7}$$

Proof of Lemma: Let G_n be the sub-σ-algebra generated by $(\hat{X}_{-n+1}^{-1},X_{-\infty}^{-n})$. Because $\hat{X}_{-i}=f^s(X_{-\infty}^{-i})$ for all i, it follows that the G_n's decrease with n and contain the sub-σ-algebra G generated by $\hat{X}_{-\infty}^{-1}$. This implies the first inequality and that $H(\hat{X}_o|\hat{X}_{-n+1}^{-1},X_{-\infty}^{-n})$ increases with n [4, page 124]. We obtain the second inequality by expanding $H(\hat{X}_{-n+1}^{\;o}|X_{-\infty}^{-n})$ ([4], page 124, property C_1), using the joint stationarity of $\{X_n,\hat{X}_n\}$ and the fact that $H(\hat{X}_o|\hat{X}_{-n+1}^{-1},X_{-\infty}^{-n})$ increases with n; that is

$$\begin{aligned} H(\hat{X}_{-n+1}^{o}|X_{-\infty}^{-n}) &= H(\hat{X}_{-n+1}|X_{-\infty}^{-n})+H(\hat{X}_{-n+2}|\hat{X}_{-n+1},X_{-\infty}^{-n})+\ldots+H(\hat{X}_o|\hat{X}_{-n+1}^{-1},X_{-\infty}^{-n}) \\ &= H(\hat{X}_o|X_{-\infty}^{-1})+H(\hat{X}_o|\hat{X}_{-1},X_{-\infty}^{-2})+\ldots+H(\hat{X}_o|\hat{X}_{-n+1}^{-1},X_{-\infty}^{-n}) \\ &\le n\; H(\hat{X}_o|\hat{X}_{-n}^{-1},X_{-\infty}^{-n+1}), \end{aligned}$$

proving the second inequality.

The quantity $\frac{1}{n}H(\hat{X}_{-n+1}^{o}|X_{-\infty}^{-n})$ will be referred to as the n^{th}-order conditional entropy. We now let $r_c^n(D)$ denote the least n^{th}-order conditional entropy attainable by any causal sliding block code with average distortion D or less, that is

$$r_c^n(D)\triangleq \inf_{\Psi(D)}\; \frac{1}{n}H(\hat{X}_{-n+1}^{o}|X_{-\infty}^{-n}). \tag{8}$$

Theorem: For a finite alphabet stationary process $\{X_n\}$ and a process $\{\hat{X}_n\}$ derived from $\{X_n\}$ by f^s, a causal sliding block code,

$$r_c^s(D)\ge r_c^n(D)\ge r_c^{n-1}(D).$$

Proof of Theorem: The first inequality follows immediately upon taking the infimums of the left and right hand sides of (7) over all causal sliding block codes with $\rho(f^{s}) \leq D$. We obtain the second inequality by first establishing that $\frac{1}{n}H(\hat{X}_{-n+1}^{o} | X_{-\infty}^{-n})$ is non-decreasing with n. By repeating steps similar to those used in the lemma, we obtain

$$H(\hat{X}_{-n+1}^{o} | X_{-\infty}^{-n}) = H(\hat{X}_{o} | \hat{X}_{-n+1}^{-1}, X_{-\infty}^{-n}) + H(\hat{X}_{-n+1}^{-1} | X_{-\infty}^{-n})$$

$$\geq \frac{1}{n}H(\hat{X}_{-n+1}^{o} | X_{-\infty}^{-n}) + H(\hat{X}_{-n+2}^{o} | X_{-\infty}^{-n+1})$$

or equivalently

$$\frac{1}{n}H(\hat{X}_{-n+1}^{o} | X_{-\infty}^{-n}) \geq \frac{1}{n-1} H(\hat{X}_{-n+2}^{o} | X_{-\infty}^{-n+1}). \tag{9}$$

Upon taking infimums of both sides of (9) over all f^{s} with $\rho(f^{s}) \leq D$ we obtain the second inequality.

Although the nth-order conditional entropy converges to the entropy rate of $\{\hat{X}_{n}\}$, an open question is does $r_{c}^{n}(D)$ converge to $r_{c}^{s}(D)$. As the following discussion demonstrates, we can lower bound $r_{c}^{n}(D)$ for all n.

LOWER BOUND TO $r_{c}^{n}(D)$

Observe that the nth-order conditional entropy and average distortion can be written in the following manner:

$$\frac{1}{n}H(\hat{X}_{-n+1}^{o} | X_{-\infty}^{-n}) = \frac{1}{n} \int H(\hat{X}_{-n+1}^{o} | X_{-\infty}^{-n} = x_{-\infty}^{-n})\, dP(x_{-\infty}^{-n}) \tag{10}$$

and

$$E\rho(X_{o}, \hat{X}_{o}) = \int E[\rho(X_{o}, \hat{X}_{o}) | X_{-\infty}^{-n} = x_{-\infty}^{-n}]\, dP(x_{-\infty}^{-n}), \tag{11}$$

where P denotes the source measure on $A_{-\infty}^{-n}$. Given any f^{s} we can partition $A_{-\infty}^{-n}$ into sets on which $H(\hat{X}_{-n+1}^{o} | X_{-\infty}^{-n} = x_{-\infty}^{-n})$ and $E[\rho(X_{o}, \hat{X}_{o}) | X_{-\infty}^{-n} = x_{-\infty}^{-n}]$ are constant. For any $z = x_{-\infty}^{-n}$, let us define $f_{z}^{s} : A_{-n+1}^{o} \to A_{-n+1}^{o}$ by

$$f_{z}^{s}(x_{-n+1}^{o}) = (f^{s}(x_{-\infty}^{-n+1}), \ldots, f^{s}(x_{-\infty}^{o})).$$

Thus f_{z}^{s} maps n-blocks into n-blocks.

Let $Q_{1}, \ldots, Q_{M}$ be an ordering of all such mappings from $A_{-n+1}^{o} \to A_{-n+1}^{o}$. Consider the partition $E(f^{s}) = \{E_{i,k}; i=1, \ldots, J; k=1, \ldots, M\}$ of $A_{-\infty}^{-n}$ where the atom $E_{i,k}$ is defined as

$$E_{i,k} \overset{\Delta}{=} \{x_{-\infty}^{-n} : x_{-n} = a_{i}, f_{z}^{s} = Q_{k}\}.$$

Because the source is Markov and $x_{-n} = a_{i}$ for all $x_{-\infty}^{-n} \varepsilon E_{i,k}$, the conditional distribution for X_{-n+1}^{o}, given $X_{-\infty}^{-n} = x_{-\infty}^{-n}$ is the same for all $x_{-\infty}^{-n} \varepsilon E_{i,k}$ and in particular it equals the conditional distribution for X_{-n+1}^{o} given $X_{-n} = a_{i}$. Additionally, $\hat{X}_{-n+1}^{o} = Q_{k}(X_{-n+1}^{o})$ when $x_{-\infty}^{-n} \varepsilon E_{i,k}$. Hence we conclude that the conditional distribution of $\hat{X}_{-n+1}^{o}$ given $X_{-\infty}^{-n} = x_{-\infty}^{-n}$ is the same for all

$x_{-\infty}^{-n}\varepsilon E_{i,k}$. Therefore

$$\frac{1}{n} H(\hat{X}_{-n+1}^{o}|X_{-\infty}^{-n}=x_{-\infty}^{-n}) \triangleq h_{i,k} \tag{12}$$

and

$$E[\rho(X_o,\hat{X}_o)|X_{-\infty}^{-n}=x_{-\infty}^{-n}] \triangleq d_{i,k} \tag{13}$$

are constants on $E_{i,k}$. For the $E(f^s)$ partition induced by f^s, there is a probability distribution $\{P_{i,k}\}$, where $P_{i,k} \triangleq Pr\{X_{-\infty}^{-n}\varepsilon E_{i,k}\}$. Using (10)-(13) we can rewrite the conditional entropy and average distortion as

$$\frac{1}{n} H(\hat{X}_{-n+1}^{o}|X_{-\infty}^{-n}) = \sum_{i=1}^{J}\sum_{k=1}^{M} h_{i,k}\cdot P_{i,k} \tag{14}$$

and

$$E\rho(X_o,\hat{X}_o) = \sum_{i=1}^{J}\sum_{k=1}^{M} d_{i,k}\cdot P_{i,k}. \tag{15}$$

Defining $\xi(D)$ as the set of all partitions $E(f^s)$ induced by causal sliding block codes f^s with $\rho(f^s)\leq D$. $r_c^n(D)$ can be rewritten as

$$r_c^n(D) = \inf_{\xi(D)} \sum_{i=1}^{J}\sum_{k=1}^{M} h_{i,k}\cdot P_{i,k}. \tag{16}$$

We must note that even for modest values of n and J, M is very large (e.g., $n=4$, $J=2$, $M>2\times 10^9$).

If the constraint set $\xi(D)$ could be described by an equivalent set of linear equalities and inequalities constraining $\{P_{i,k}\}$, then (16) could be solved by linear programming. At the present we lack such a characterization for $\xi(D)$. We have, however, derived a set of linear constraints that the distribution $\{P_{i,k}\}$ of $E(f^s)$ must satisfy. These constraints are summarized below.

(i) For $\{P_{i,k}\}$ to be a probability distribution we must have $P_{i,k}\geq 0$ for all (i,k) and $\sum_{i=1}^{J}\sum_{k=1}^{M} P_{i,k}=1$.

(ii) Using (15) the distortion constraint, $\rho(f^s)\leq D$, becomes

$$\rho(f^s) = \sum_{i=1}^{J}\sum_{k=1}^{M} d_{i,k}\cdot P_{i,k}\leq D.$$

(iii) For any $y=x_{-\infty}^{-1}$ let $f_y^s: A\to A$ be defined according to $f_y^s(x_o)=f^s(x_{-\infty}^{o})$ and let $R_1,\ldots,R_j$ be an ordering of all such mappings. Now consider the sets

$$F_{\ell,m,q} \triangleq \{y=x_{-\infty}^{-\ell-1}: x_{-\ell-1}=a_m, f_y^s=R_q\},$$

where $\ell=0,1,\ldots,n-1$, $m=1,\ldots,J$ and $q=1,\ldots,J^J$. Because f^s is stationary, the probability of the sets $F_{\ell,m,q}$ must not depend on ℓ. Now we observe that if $\ell=0,\ldots,n-2$, then

$$F_{\ell,m,q} = \bigcup_{i=1}^{J} \bigcup_{k=1}^{M} \{E_{i,k} \times (G_{\ell,k,q} \cap P_m^{-\ell-1})\} \tag{17}$$

where $G_{\ell,k,q} \triangleq \{x_{-n+1}^{-\ell-1} : Q_k(x_{-n+1}^{-\ell}) = R_q(x_{-\ell})$ for all $x_{-\ell}\}$ and $P_m^{-\ell-1} \triangleq \{x_{-n+1}^{-\ell-1} : x_{-\ell-1} = a_m\}$. If $\ell=n-1$, then

$$F_{\ell,m,q} = \bigcup_{k} E_{m,k}, \tag{18}$$

where the union is taken over all k such that $Q_k(x_{-n+1}) = R_q(x_{-n+1})$ for all x_{-n+1}. We now show that $\Pr\{F_{\ell,m,q}\}$ can be evaluated in terms of the $\{P_{i,k}\}$ and the source transition probabilities. In particular, if $\ell=0,\ldots,n-2$

$$\begin{aligned} \Pr\{F_{\ell,m,q}\} &= \sum_{i=1}^{J} \sum_{k=1}^{M} \Pr\{E_{i,k}\} \cdot \Pr\{G_{\ell,k,q} \cap P_m^{-\ell-1} \mid E_{i,k}\} \\ &= \sum_{i=1}^{J} \sum_{k=1}^{M} P_{i,k} \cdot \Pr\{G_{\ell,k,q} \cap P_m^{-\ell-1} \mid X_{-n} = a_i\}. \end{aligned} \tag{19}$$

If $\ell=n-1$,

$$\Pr\{F_{\ell,m,q}\} = \sum_{k} P_{m,k} \tag{20}$$

where the sum is taken over the same set as the union in (18). Note that the quantities $\Pr\{G_{\ell,k,q} \cap P_m^{-\ell-1} \mid X_{-n} = a_i\}$ can be evaluated in terms of the source transition probabilities. We can now see that for each (m,q), it is possible to express the stationarity constraint using (n-1) linear constraints on $\{P_{i,k}\}$ of the following form

$$0 = \Pr\{F_{\ell,m,q}\} - \Pr\{F_{\ell+1,m,q}\} = \sum_{i,k} a_{i,k}(\ell,m,q) \cdot P_{i,k}, \quad \ell = 0,1,\ldots,n-2.$$

In summary we have one linear constraint from (i), one linear constraint from (ii) and $J \times J^J \times (n-1)$ linear constraints from (iii). Let $\Omega(D)$ denote the set of all distributions $\{P_{i,k}\}$, satisfying the derived linear constraints. We now have, simply by construction, if f^s belongs to $\Psi(D)$, then there exists a partition $E(f^s)$ with the distribution $\{P_{i,k}\}$ belonging to $\Omega(D)$. Because the constraint set $\Omega(D)$ contains all such distributions induced by f^s, the minimization of $\sum_{i=1}^{J} \sum_{k=1}^{M} h_{i,k} \cdot P_{i,k}$ over $\Omega(D)$ can be no larger than the minimization (16). This proves the following result.

Theorem: For any Markov source

$$r_c^n(D) \geq \inf_{\Omega(D)} \sum_{i=1}^{J} \sum_{k=1}^{M} h_{i,k} \cdot P_{i,k} \triangleq r_c^{LP}(D). \tag{21}$$

The infimum in (21) is actually achieved by some $\{P_{i,k}\}$ because

$\sum_{i=1}^{J} \sum_{k=1}^{M} h_{i,k} \cdot P_{i,k}$ is continuous and $\Omega(D)$ is compact. Given $\{P_{i,k}\} \varepsilon \Omega(D)$, we cannot always find an $f^s \varepsilon \Psi(D)$ that could induce this distribution. We have been able to prove, for certain sources and distortions, there exist $f^s \varepsilon \Psi(D)$, whose performances are arbitrarily close to $r_c^{LP}(D)$. The proof involves constructing codes using a specialization of Rohlin's Theorem [5] to partitions measurable with respect to the sub-σ-algebras $A_{-\infty}^{-n}$. For such cases (21) is an equality. We conjecture this so more generally. In any event, the right hand side of (21) is a lower bound to that can be evaluated by linear programming.

EVALUATION OF BOUNDS TO $r_c^s(D)$

A binary symmetric Markov Source with transition probability ($\Pr\{X_n=0|X_{n-1}=1\} = \Pr\{X_n=1|X_{n-1}=0\}$) one quarter has been selected for comparing the upper and lower bounds to $r_c^s(D)$. The Hamming metric was used for the distortion measure, which means the average distortion, $E\rho(X_o, f^s(X_{-\infty}^o))$, for any f^s is $\Pr\{X_o \neq \hat{X}_o\}$. In Figure 1 we plot $r_c^{LP}(D)$ for n=1,2. It is also possible to evaluate $r_c^{LP}(D)$ for n=3, but as mentioned earlier, the distribution for n=4 has more than two billion terms. We have also included, for comparison, the best known bounds to the rate distortion function for this source [6]. These bounds to the rate distortion function, which are tight for small distortions, indicate for a given reproduction entropy rate, that causal codes will have significantly higher distortions. See Figure 1.

REFERENCES

[1] D. L. Neuhoff, "Causal Source Coding," Proceedings Sixteenth Annual Allerton Conference, Oct. 1978, pp. 735-743.

[2] D. L. Neuhoff and R. K. Gilbert, "Causal Source Codes," to appear in IEEE Trans. on Inform. Theory.

[3] P. C. Shields and D. L. Neuhoff, "Block and Sliding Block Source Coding," IEEE Trans. on Inform. Theory, Vol. IT-23, March 1977, pp. 211-215.

[4] P. Billingsley, Ergodic Theory and Information, New York: Wiley, 1965.

[5] P. C. Shields, The Theory of Bernoulli Shifts, Chicago: University of Chicago Press, 1973.

[6] T. Berger, "Explicit Bounds to R(D) for a Binary Symmetric Markov Source," IEEE Trans. on Inform Theory, Vol. IT-23, Jan. 1977, pp. 52-59.

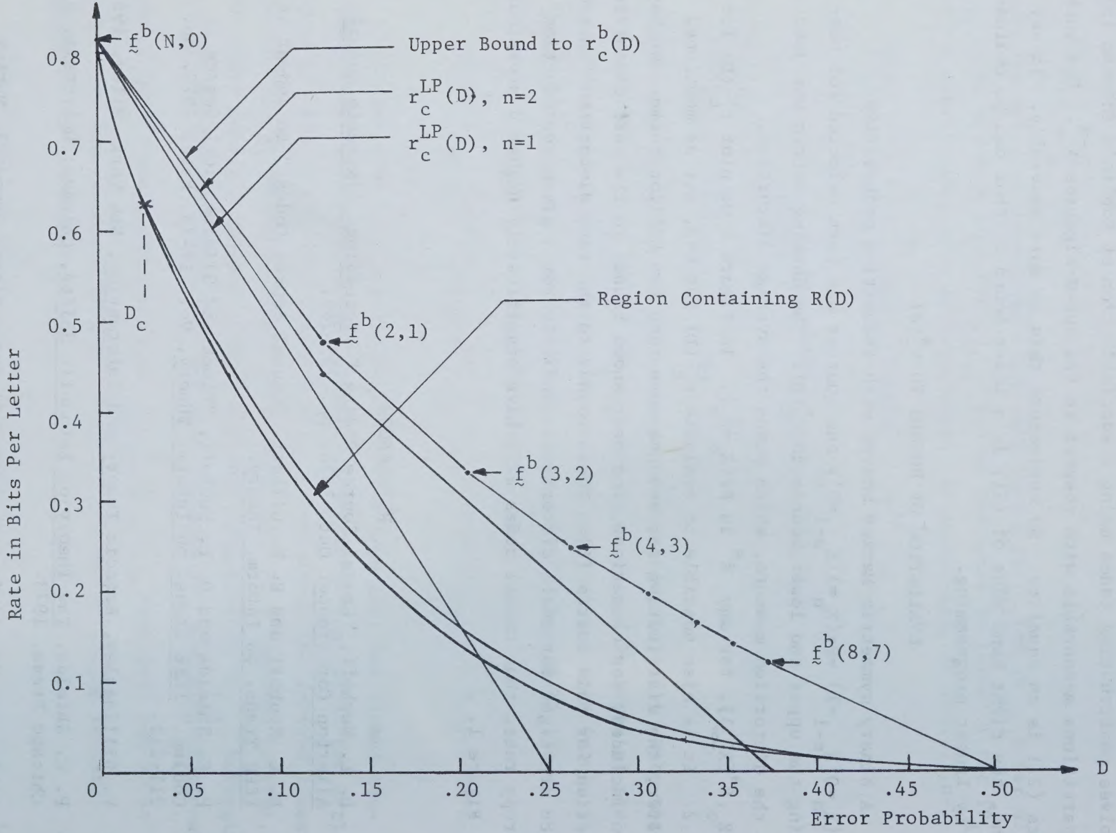


Figure 1. Bounds to $r_c^s(D)$

A UNIVERSAL CODE OF REDUCED COMPLEXITY FOR THE BINARY SOURCE

K. M. MACKENTHUN, JR.
College of Engineering
University of South Carolina
Columbia, SC 29208

Recently, Gallager has given the construction for a run-length code which is an optimal code for a source on the nonnegative integers with a geometric probability assignment. We show that Gallager's code may be modified to perform as a universal code for the memoryless binary source. This code construction has several important practical advantages over the Lynch-Schwalkijk-Davisson universal code for the memoryless binary source.[1]

The construction of Gallager's code is as follows: let $P(i)=(1-\theta)\theta^i$ be a probability assignment on the set of nonnegative integers where θ is an arbitrary real number, $0<\theta<1$. Let ℓ be the integer satisfying $\theta^\ell+\theta^{\ell+1} \leq 1 < \theta^\ell+\theta^{\ell-1}$ and represent each nonnegative integer i as $i = \ell j+r$ when j is the integer part of i/ℓ and $r = i \bmod \ell$. Encode j by j zeros followed by a single one, and encode r by a Huffman code, using codewords of length $\lfloor \log_2\ell \rfloor$, for $r < 2^{\lfloor \log\ell+1 \rfloor} - \ell$, and length $\lfloor \log_2\ell \rfloor + 1$ otherwise. An optimal code is the concatenation of these two codes. Note that this code has a very simple encoding and decoding procedure. The encoder simply counts up to ℓ zeros from the source and then produces a single zero at the output. When a one appears from the source, the encoder produces a one, terminating the unary code, and then produces the codeword for the position of the incoming one within the subblock of ℓ digits. When using this code on a succession of source outputs, one is naturally interested in the redundancy per bit of the source code. It may be shown that this is approximately 1% of the source entropy per bit, $H(\theta)$, for all values of θ.

The construction of the universal run-length code is as follows. Suppose a sequence of n bits having k ones is observed. In the first part of the code word, send the number of ones using a fixed-length code of at most $\lceil \log n \rceil$ bits, as in the Lynch et. al. construction. The value of k/n empirically determines a θ, which can be used to determine the appropriate ℓ. Using this ℓ, encode the n-bit sequence via Gallager's code construction. As $n \rightarrow \infty$, $k/n \rightarrow \theta$ a. s. and the coding rate of the second part converges to $H(\theta)$ (within 1%). Since the coding rate of the first part goes to 0, this code is an optimal universal code. Note that the run-length universal code has an important practical advantage over the Lynch et. al. universal code; as n becomes large, the complexity of the run-length code does not increase while the encoding and decoding complexity of the Lynch code increases exponentially because of the encoding of $\binom{n}{k}$.

In a practical situation, one would not let n grow without bound. One would segment the string of source outputs into blocks of length n and encode each block using either of the two universal codes above. It is clear that such codes are not optimal since there is a fixed overhead of approximately $\log n/n$ bits per block. Using this format, the run-length scheme may be configured into a non-block memoryless or adaptive form. In this form, the first part of each codeword is used to send θ instead of k; the empirical value of θ is recursively updated at each block using the θ of the previous block and the number of ones in the previous block. Note that in this scheme we use the θ determined by all past blocks to encode the current block, and therefore we can send the current block with no delay, in contrast to the Lynch code where the delay is proportional to n.

1. L. D. Davisson, "Universal noiseless coding," IEEE Trans. Inform. Theory, vol. IT-19, Nov. 1973, pp. 783-795.

This work was supported in part by NSF Grant ENG79-08073.

VARIABLE RATE ENCODING USING ADAPTIVE DELTA MODULATION*

P.P. PREZAS and J.L. LoCICERO
Illinois Institute of Technology
Department of Electrical Engineering
Chicago, Illinois 60616

ABSTRACT

In this paper we present a variable sampling frequency adaptive delta modulation (ADM) system intended for audio signal encoding. The objective is to reduce the channel bit rate required by other delta modulation systems, while maintaining the same quality of signal reproduction. The ADM rate is selected by first measuring the slope during a time segment of the audio signal. The slope measure is mapped to an ADM rate via a set of signal-to-noise (SNR) curves and a specified performance level. We present a probabilistic analysis of the slope and compare it with simulation results. Using a composite test signal to simulate a voice waveform, the system performance has been studied as a function of the observation time segment. SNR curves are presented and it is shown that a bandwidth compression of at least 30% can be achieved over a conventional ADM encoder.

INTRODUCTION

Delta modulation is a technique of digital source encoding by which an analog signal is transformed into a sequence of binary digits (bits) for transmission thru a channel. At the receiver, the bits are decoded to reconstruct an estimate of the original analog signal. The block diagram of an ADM encoder is shown in Fig. 1. In this system, the analog input, $x(t)$, assumed bandlimited to f_m, is compared with an estimate signal, $\hat{x}(kT_s)$. If the result of the comparison, $\xi_x(kT_s)$, is positive, the output, $e_x(kT_s)$, is +1. If $\xi_x(kT_s)$ is negative, the output is -1. In every clock period, $T_s = 1/f_s$, the estimate is incremented by an amount $S_x(kT_s)$ determined by the step size algorithm.

In this system, the ADM clock (or sampling) frequency is well above the Nyquist rate, i.e., $f_s >> 2f_m$. A mathematical description of this ADM system is given by the following set of equations:

$$e_x(kT_s) = \text{sgn}[\xi_x(kT_s)], \tag{1}$$

where

$$\xi_x(kT_s) = x(kT_s) - \hat{x}(kT_s) \tag{2}$$

and

$$\hat{x}(kT_s) = \hat{x}[(k-1)T_s] + S_x(kT_s). \tag{3}$$

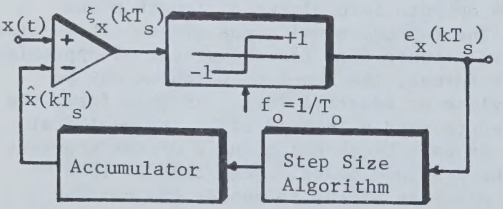


Fig. 1. DM Encoder

There are many step size algorithms that can be employed for $S_x(kT_s)$. We shall use the Song audio mode algorithm [1,2], where the step size increases linearly, i.e.,

$$S_x(kT_s) = |S_x[(k-1)T_s]|e_x[(k-1)T_s] + Se_x[(k-2)T_s]. \tag{4}$$

Here S is the magnitude of the minimum step size. The ADM decoder is just the feedback portion of Fig. 1 followed by a low pass filter.

We point out that the ADM sampling frequency of this conventional system is constant. We propose to vary the ADM clock rate, using the same step size algorithm, in an effort to reduce the net channel rate while still maintaining the same quality of signal reproduction.

---

*Portions of this research were supported by the Engineering Foundation grant RC-A-77-1D.

A VARIABLE SAMPLING FREQUENCY ADM SYSTEM

The Basic Concept of Variable Rate Encoding

The ADM as a digital source encoder is particularly sensitive to slope changes in the input signal. To quantitatively see the dependence on slope, consider a large change in the amplitude of the input signal over a specific time interval. If the ADM is to follow the input signal without slope-overload, the encoder must produce a sufficient number of bits to make tracking possible, necessitating a high sampling rate. What we observe is a direct relationship between the sampling rate of the ADM and the slope of the input signal. Likewise, we note that the performance of the encoder largely depends upon the way it responds to slope changes of the input signal in attempting to minimize slope-overload.

The basic principle behind varying the ADM sampling rate lies in the fact that the slope of a time segment of an audio signal varies continually, but remains within a finite range. Therefore, we should be able to adjust the ADM clock rate to have a desired performance (SNR) for that particular time segment. In addition, practice has shown [3] that for conversational-type audio signals, the channel is unused almost 50% of the time due to pauses interlaced between speech phrases.

Based on the above observations, we propose a variable sampling rate system, shown diagrammatically in Fig. 2. The operation of this system can be understood by tracing thru its blocks. First, we observe the analog input, x(t), for a T second time interval and take a measure of its slope, m(kT). For the moment we consider T as a parameter of our system. We shall focus on it later. Prior to encoding, the total range of the slope of the input signal has been divided into segments. Depending on the slope segment in which m(kT) happens to fall, the frequency selector assigns a particular sampling rate, $f_s(kT)$, to the ADM. Later in this section we shall detail the mapping of slope to ADM sampling rate.

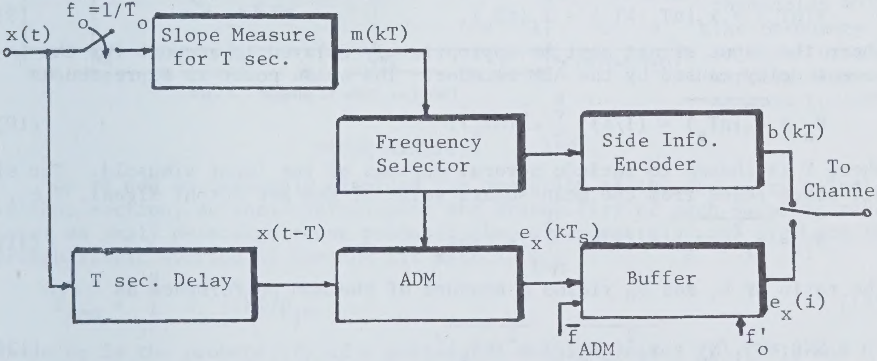


Fig. 2. Variable Bit Rate Adaptive Deltamodulation System.

After the sampling frequency has been set, the ADM encodes the input signal which has been delayed by T seconds. The bits, $e_x(kT_s)$, coming out of the ADM encoder, are combined in an output buffer with the side information bits, b(kT), and transmitted thru the channel. The side information bits are needed to inform the receiver exactly which sampling frequency is being used in the transmitter during every T second time interval. The entire process is repeated on the input signal every T second time interval.

The slope measure is obtained by sampling the input signal at a very high rate, $f_o = 1/T_o$, such that

$$T/T_o = N, \tag{5}$$

where N is an integer. The difference signal,

$$\Delta(nT_o) = x(nT_o) - x(nT_o - T_o), \tag{6}$$

is then used to determine a slope measure of the input. Of the many slope measures that can be considered, in this paper, we shall use the maximum absolute value of the slope during the T second interval, that is,

$$m(kT) = [|\Delta(nT_o)|/T_o]_{max} \quad \text{for } 1 \le n \le N. \tag{7}$$

The calculated slope measure is then used to select a sampling frequency for the ADM. The technique employed for this selection is detailed next.

ADM Rate Selection

The frequency selector shown in Fig. 2 maps the slope measure to an ADM sampling rate based on a set of performance curves. As a measure of performance, we shall use the standard SNR employed for audio signals. These curves are the result of a digital computer simulation using the system shown in Fig. 3. The input is of the form

$$x(t) = \cos(2\pi f_i t), \tag{8}$$

where f_i takes on values between 300 and 3300 Hz. The slope range of x(t) obviously depends upon the range of f_i. The output of the ADM encoder shown in Fig. 3 is applied to a fourth-order digital band pass filter (BPF) yielding $\hat{x}_f(nT_s)$. The input signal is also sampled at the ADM rate, f_s, and filtered, thru an identical digital BPF used on the estimate, yielding $x_f(nT_s)$. In this way, the effect of the BPF is the same on both the input signal and the estimate.

The noise power is computed as the mean-square value of the error signal,

$$\varepsilon(nT_s) = x_f(nT_s - \lambda T_s) - \hat{x}_f(nT_s), \tag{9}$$

where the input signal must be appropriately delayed to account for the inherent delay caused by the ADM encoder. The noise power is expressed as

$$N_o = \overline{\varepsilon^2(nT_s)} = (1/K) \sum_{n=1}^{K} \varepsilon^2(nT_s), \tag{10}$$

where K is chosen to include several periods of the input sinusoid. The signal power comes from the mean-square value of the BPF output signal, i.e.,

$$S_i = \overline{x_f^2(nT_s)} = (1/K) \sum_{n=1}^{K} x_f^2(nT_s). \tag{11}$$

The ratio of S_i and N_o yields a measure of the ADM performance as

$$SNR = S_i/N_o = \overline{x_f^2(nT_s)} \; / \; \overline{\varepsilon^2(nT_s)}. \tag{12}$$

The simulation described above was performed for several values of f_s. Figure 4 shows a plot of SNR (in dB) versus the maximum slope in the sinusoid for a family of f_s values between 8 kilo-Hz and 64 kilo-Hz. From this plot, it is obvious that, for a given ADM rate, as the slope of the input increases, the SNR decreases. We also notice that, for a given value of slope, the SNR is directly proportional to the ADM rate. Both observations are expected results.

These performance curves will now be used as the basis of our frequency

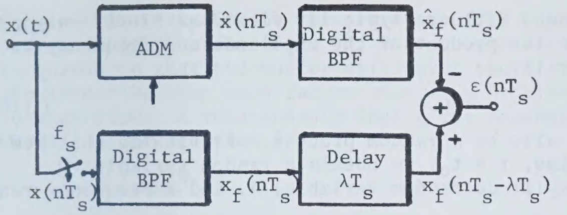


Fig. 3. System to Measure ADM Performance.

selector network in an effort to force the variable rate ADM system to maintain a desired SNR. Suppose we let the desired SNR be H (dB) and draw a horizontal line on the performance curves at this level. We shall have then established a threshold from which we can determine the ADM clock rate. For every T second time interval, $f_s(kT)$ will now be chosen according to the following rule:

> "For the calculated slope measure, m(kT), choose the minimum value of ADM clock rate, $f_s(kT)$, to guarantee that SNR $\geq$ H".

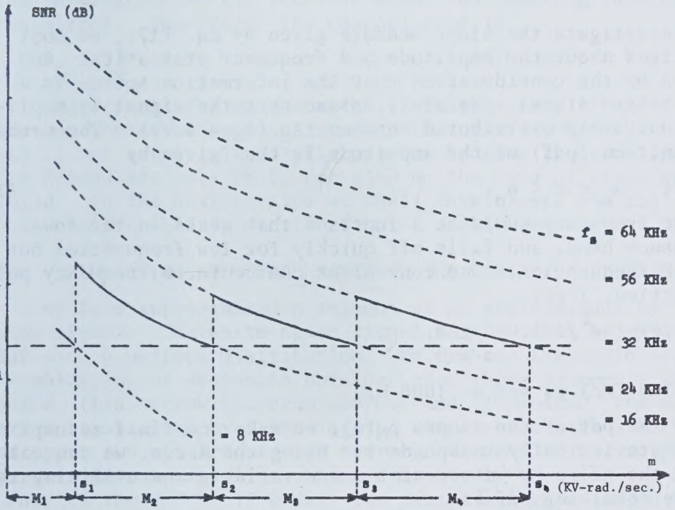


Fig. 4. Mapping of m(kT) to f_s(kT)

Using this rule, we divide the slope axis into a set of intervals, $\{M_1, M_2, \cdots\}$ as shown in Fig. 4, and assign to each interval an ADM rate corresponding to the solid SNR curve in that interval. The values of the maximum slope $\{s_1, s_2 \cdots\}$ between the intervals are the thresholds which the frequency selector uses in mapping slope measure to ADM rate.

SLOPE ANALYSIS

Up to now we have only examined the operation of the frequency selector. In this section, we shall investigate the probability of each sampling rate. Later we shall determine these probabilities experimentally and evaluate the probabilistic average of the ADM bit rate, i.e.,

$$\overline{f}_{ADM} = \sum_{j=1}^{R} f_{s,j}(kT)p_j, \tag{13}$$

Where p_j is the probability of a particular sampling rate, $f_{s,j}(kT)$, and R is the total number of different ADM rates.

In performing the theoretical analysis, we initially model the analog input as an amplitude and frequency modulated signal, i.e.,

$$x(t) = A(t)\cos[\phi(t)], \tag{14}$$

where A(t) is the amplitude variation and $\phi(t)$ is the phase variation. The instantaneous frequency (in rad./sec.) is given by

$$\Omega(t) = d\phi/dt. \tag{15}$$

In this type of model, $A(t)$ and $\Omega(t)$ are typically viewed as slowly varying random processes. If we use the product of the amplitude and frequency as a measure of the slope, that is,

$$M(t) = A(t)\Omega(t), \tag{16}$$

then the slope measure will also be a random process. By viewing this new random process at a given time, $t = t_o$, we obtain a random variable, M, which is a function of an amplitude random variable, A, and a frequency random variable, Ω, that is,

$$M = A\Omega. \tag{17}$$

If we were to assume for a moment that the input takes the simplest possible form in the time domain, that of a sinusoid, such that

$$x(t) = a_1 \sin(\omega_1 t), \tag{18}$$

then the maximum slope is the product of the amplitude and the radian frequency, $a_1\omega_1$.

To further investigate the slope measure given by Eq. (17), we must make some assumptions about the amplitude and frequency statistics. Our choice is governed by the consideration that the information source is a bandlimited speech-type signal. We shall assume that the signal is amplitude limited and uniformly distributed between $\pm A_o$ (A_o = 1 V.). The probability density function (pdf) of the amplitude is thus given by

$$p_A(a) = A_o/2, \quad -A_o \leq a \leq A_o. \tag{19}$$

A good pdf for the frequency would be a function that peaks in the lower part of the frequency band, and falls off quickly for low frequencies but gradually for high frequencies. One convenient choice for a frequency pdf is a Rayleigh function, i.e.,

$$\begin{aligned} p_\Omega(\omega) &= (\omega/c)\exp(-\omega^2/2c), & \omega \geq 0, \\ &= 0, & \omega < 0, \end{aligned} \tag{20}$$

where the peak value ($\sqrt{c}$) is set at 1000 Hz.

To calculate the pdf of the slope, $p_M(m)$, we make one final assumption, i.e., A and Ω are statistically independent. Using the above, we can calculate the pdf of the slope by introducint a new variable and utilizing the Jacobian [4]. The resulting pdf is

$$p_M(m) = (\sqrt{\pi}/2A_o\sqrt{2c})[1 - \mathrm{erf}(|m|/A_o\sqrt{2c})]. \tag{21}$$

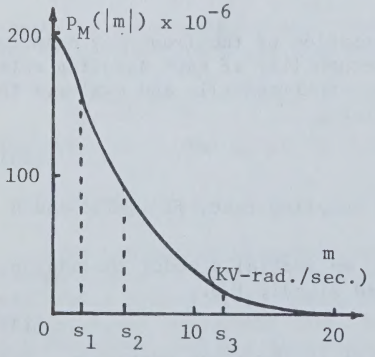


Fig. 5. Pdf of $|m|$.

Figure 5 is a plot of $p_M(|m|)$ based on Eq. 21. To determine the probability of a particular rate, p_j, all we must do is compute the probability that the maximum slope of $x(t)$ lies within a specific range, i.e.,

$$p_j = \text{Prob.}(s_{j-1} \leq m \leq s_j), \tag{22}$$

where s_{j-1} and s_j are the slope thresholds shown in Fig. 4. This probability is just the area under the slope pdf between the limits set by the thresholds, that is,

$$p_j = \int_{s_{j-1}}^{s_j} p_M(|m|)dm. \tag{23}$$

Using Eqs. (13) and (23) we can now calculate the average ADM rate,

$\overline{f}_{ADM}$. This will also be the average buffer input rate (see Fig. 2). The average output buffer rate, f_o', must be greater than the average input rate to guarantee that the buffer will never overflow. On the other hand, f_o' should not be very much larger than $\overline{f}_{ADM}$ or else we shall cause the buffer to underflow. A relationship that gives reasonable probabilities for buffer underflow and overflow is

$$f_o' = \overline{f}_{ADM}/0.98. \tag{24}$$

FINAL CHANNEL BIT RATE

To relate the final channel rate, f_R, to the average output buffer rate, we must consider the multiplexing operation between the side information encoder and the buffer. Since we have R sampling rates, we need

$$b = R \text{ [modulo 2]} \tag{25}$$

additional bits, at the beginning of every T second time interval, to provide information to the receiver about the sampling rate being used in the transmitter. Therefore, the channel rate is

$$f_R = f_o' + b/T. \tag{26}$$

The effect of T on the side information rate is clear from the above equation. What we have not studied thus far is the effect of T on $\overline{f}_{ADM}$. This is not a trivial analytic problem because the sampling rate probabilities depend not only on T, but also on the type of slope measure being employed. In the next section we shall investigate how the length of T affects the probability distribution of the ADM rates.

SIMULATION AND PERFORMANCE OF THE VARIABLE BIT RATE ADM

We have approximated a segment of an audio signal by filtering a white noise signal. The white noise signal is simulated by a random number generator with a uniform distribution. We created the audio signal segment with a combination of sequences obtained from three sources: (a)a wide-spectrum source, (b)a narrow-spectrum source, and (c)pauses. The wide spectrum source is obtained by filtering the white noise with a BPF that has lower and upper cut-off frequencies set at 300 and 3300 Hz, respectively. The narrow-spectrum signal is obtained by merely changing the upper cut-off frequency of the BPF to 1000 Hz. The pauses are represented by a zero amplitude segment and they are used to indicate that in a typical conversation, the channel is unused almost 50% of the time. Figure 6 shows the arrangement used. It is the concatination of several 100 msec. segments. The W and the N denote wide- and narrow-spectrum signals of full amplitude while the W/2 and the N/2 indicate that the signals are limited to only half the maximum allowable amplitude, i.e., ±0.5 V. This will be used as a test signal for our variable rate ADM system. Figure 7 shows the slope distribution obtained from this test signal. A comparison of this curve with Fig. 5 (analytic pdf) indicates reasonable agreement.

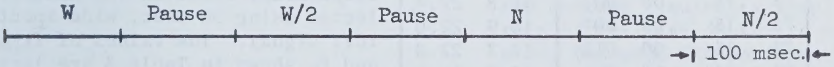


Fig. 6. Source Arrangement for the Test Signal.

To study the time interval T, we must return to the performance curves (Fig. 4). As an example, let the desired performance of our system be 20 dB. A horizontal line thru the SNR curves at the 20 dB level defines the slope thresholds for sampling rates of 16, 24, 32 and 56 KHz according to Table 1. for a practical design, the minimum observation time, T_{min}, must allow all

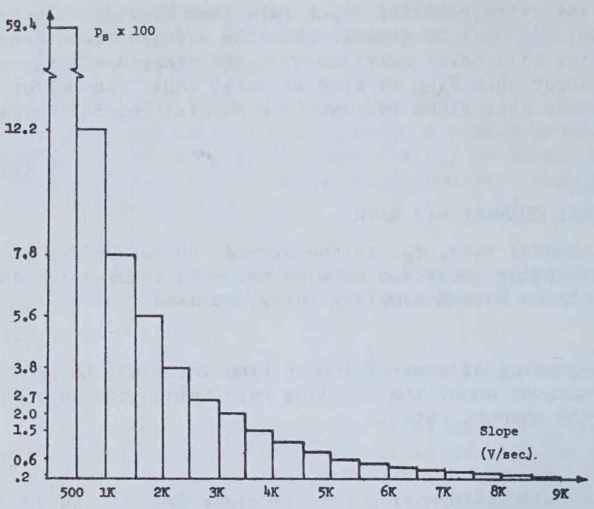


Fig. 7. Slope Histogram of the Test Signal.

of these rates to complete an integer number of cycles. In this case, T_{min} = (1/8000) sec. Now T can be, in general, any multiple of T_{min}, i.e.,

$$T = nT_{min}, \qquad (27)$$

where n is a positive integer.

One of the objectives of this simulation was to gather probabilistic data on the sampling rates of the ADM encoder as T varies. We have allowed n, in Eq. (27), to assume values of 1 to 10. The input to the system was the test signal described previously. The results are presented in Table 2. We observe that as T increases, the probability that the encoder will use the minimum sampling rate decreases, while the probability of using the maximum rate increases. This is expected since as the observation time increases, it becomes more likely for the maximum slope measure to produce a large value. This table also shows the average ADM rate and the final channel rate, calculated using Eqs. (13), (24) and (26). In this example, we need two bits of side information to specify the four sampling rates. As n increases, the effect of the side information bits on f_R obviously decreases.

Table 1. Mapping of Maximum Slope Measure to ADM Rate (20 dB SNR).

m(kT), (V/sec.)	(0 - 3140)	(3140 - 6280)	(6280 - 10990)	(>10990)
$f_s(kT)$, (KHz)	16	24	32	56

Table 2. Probability Distribution of Sampling Rates, Average ADM Rate and Final Channel Rate.

$\frac{T}{T_{min}}$	Rate Probability, p_i				$\bar{f}_{ADM}$	f_R
	f_s (KHz)				(KHz)	
	16	24	32	56		
1	.84	.11	.03	.002	17.3	33.6
2	.80	.13	.05	.003	18.0	26.3
3	.78	.14	.06	.005	18.5	24.2
4	.76	.15	.07	.007	18.7	23.1
5	.75	.15	.08	.007	18.8	22.4
6	.73	.15	.09	.007	18.9	22.0
7	.73	.15	.09	.013	19.3	22.0
8	.72	.16	.10	.016	19.5	21.9
9	.71	.16	.10	.013	19.5	21.7
10	.70	.16	.11	.018	19.8	21.8

Next we evaluate the overall performance of our system and study its dependence on T. To evaluate performance, we use a 30 msec. segment of the wide-spectrum signal as an input. The results are plotted in Fig. 8. We observe that the SNR increases with T due to the higher sampling rates making tracking of the input signal easier. Table 3 shows $\bar{f}_{ADM}$ and f_R, computed temporally, based on the date collected using 30 msec. wide-spectrum test signal. The values of $\bar{f}_{ADM}$ and f_R shown in Table 3 are larger than those computed for Table 2 due to the short duration of the 30 msec. wide-spectrum test signal in comparison to the composite signal used to develop the probabilistic results. Table 3 will be used only for comparison of the system performance with the performance of a conventional ADM encoder. To conduct this comparison, we have evaluated the SNR of the conventional ADM for several sampling rates between 16 KHz and 64 KHz, using the wide-spectrum test signal. The results are plotted in Fig. 9.

Table 3. Average ADM Rate and Final Channel Rate from Temporal Data.

T/T_{min}	1	2	3	4	5	6	7	8	9	10
$\bar{f}_{ADM}$, (KHz)	25.1	27.1	28.0	28.5	29.7	31.5	32.1	31.6	32.7	33.7
f_R, (KHz)	41.6	35.6	33.9	33.1	33.5	34.8	35.0	34.3	35.1	36.0

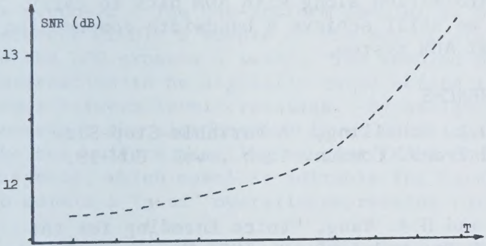


Fig. 8. Performance Of the Variable Sampling Rate ADM.

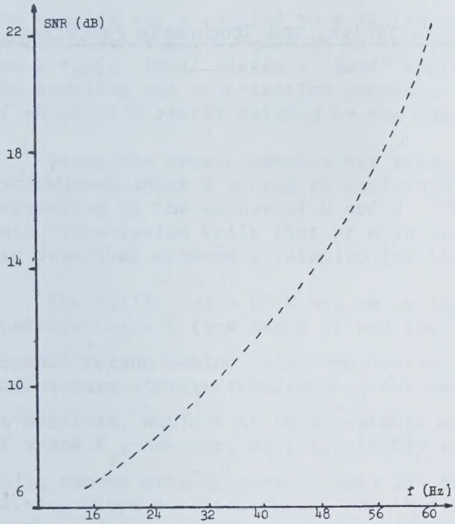


Fig. 9. Performance of a Conventional ADM.

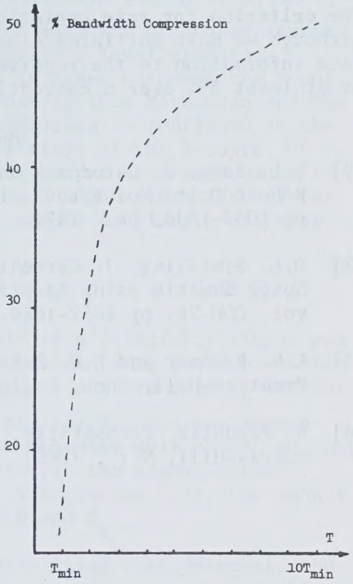


Fig. 10. Bandwidth Compression vs. Observation Period.

As an example of system comparison, let's consider the case where the variable sampling rate ADM exhibits a 12 dB SNR. From Fig. 8 we note that the required length of T is at least $5T_{min}$. Referring to Table 2, the probabalistic $\bar{f}_{ADM}$ and f_R are 18.8 KHz and 22.4 KHz respectively; while from Table 3, the temporal f_{ADM} and f_R are 29.7 KHz and 33.5 KHz. Figure 9 on the other hand shows that a conventional ADM encoder requires a sampling rate of at least 40 KHz to exhibit the same SNR performance. Consequently, using the probabilistic results, we achieve a 44% reduction in channel rate with the variable rate system. If we look at the temporal results, the reduction is about 16%. Of course, the probabilistic results are more realistic since the characteristics of an actual speech waveform are more similar to the composite signal rather than the 30 msec. wide-spectrum signal.

Our system comparison can be broadened to include all the probabilistic results displayed in Table 2. Figure 10 is a plot of the channel bandwidth compression achieved when the proposed variable rate system is used instead of the conventional ADM. We see that a bandwidth compression of at least 30% can be achieved over the conventional ADM encoder.

CONCLUSIONS

We have introduced a novel modification to the conventional ADM. Our variable rate system allows both the step size and the clock rate to vary. The criterion for rate variation is a measure of the input signal slope. Although we must multiplex side information along with ADM bits to carry rate information to the receiver, we still achieve a bandwidth compression of at least 30% over a conventional ADM system.

REFERENCES

[1] C.L. Song, J. Garodnick and D.L. Schilling, "A Variable-Step-Size Robust Delta Modulator", IEEE Trans. Commun. Tech., vol. COM-19, pp.1033-1044, Dec. 1971.

[2] D.L. Schilling, J. Garodnick and H.A. Vang, "Voice Encoding for the Space Shuttle Using Adaptive Delta Modulation", IEEE Trans. on Commun., vol. COM-26, pp.1652-1659, Nov. 1978 (Part I).

[3] L.R. Rabiner and R.W. Schafer, Digital Processing of Speech Signals, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1978.

[4] A. Papoulis, Probability, Random Variables, and Stochastic Processes, McGraw-Hill, N.Y., 1965.

A NON-UNIFORM SAMPLING APPROACH TO PREDICTIVE CODING*

JON W. MARK and TERENCE D. TODD
Department of Electrical Engineering
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1

ABSTRACT

A non-uniform sampling approach to digital encoding of analog sources is proposed. The non-uniform sampler is basically a level crossing detector (LCD); a sample of the source signal is taken whenever the input to the LCD crosses a level. The sampled process is thus Markovian and the information to be digitally coded before transmission is contained in the times between level crossings. By assigning strings of the 2-tuple 00 to represent the time between level crossings, the output binary sequence of the non-uniform sampling encoder (NSE) contains a high probability of the 0 symbol, which makes it suitable for further simple run-length encoding to attain a "good" overall compression ratio.

1. INTRODUCTION

Differential pulse code modulation (DPCM)[1] is a relatively simple and practical data compression scheme. Data compression takes place by virtue of the fact that the source signal is sampled at a uniform rate f_s and that the samples are quantized to a finite number of, say N, bits per source sample. For a given value of N, the quantizer stepsize d is chosen so that the 2-tuple (N,d) yields a "good" approximation to the signal dynamic. The sampling and quantization processes incur information loss, the amount of which is directly related to the values of N and f_s.

Since the source samples are separated by a fixed time interval, the information about a source is contained in the quantized levels, which are determined by the values of N and d. The value of N determines the rate of data transmission while that of d influences the signal distortion. Max[12] has described optimum strategies for the selection of N and d.

The ability of a DPCM system to follow a source signal depends on the quantizer size N (and hence d) and the sampling frequency f_s. Since the primary reason behind data compression is to reduce the transmission rate and/or data storage requirement, one would like to make N and f_s as small as possible, subject to an acceptable distortion tolerance. Small values of N and f_s, however, will inevitably result in a large signal distortion, which can be grossly grouped into i) slope overload noise and ii) granular noise. Slope overload occurs when the input signal changes faster than the DPCM can follow, while granular noise is proportional to the quantizer stepsize d. Thus, for sources which have intermittent occurrences of fast changing and relatively quiescent intervals, slope overload noise and granular noise will occur in distinct regions. To minimize the effect of

---

*This work has been supported by the Natural Sciences and Engineering Council of Canada under Grant A7779 and a Postgraduate Scholarship.

[1] A delta modulation (DM) system can be viewed as a special case of DPCM with N=1 bit. In this paper reference made to DPCM will also include DM as a special case.

slope overload and granular noise, it seems reasonable to sample the fast changing regions at a higher rate than the slow changing regions, as well as assigning different quantizer stepsizes. Adaptive delta modulation (ADM) samples the source at a fixed frequency and assigns varying stepsizes to attain the best compromise in signal distortion. It is apparent, however, that non-uniform sampling can enhance the signal following capability of an encoder.

An asynchronous delta modulation scheme [8] sends a pulse whenever the source signal crosses a threshold level. In this way more pulses are sent during fast changing intervals and fewer pulses during relatively quiescent intervals, i.e., the signal is sampled non-uniformly. If the quiescent intervals are long and the number of these long intervals are large, then the average number of pulses transmitted would be relatively small. Yet, the transmitted pulses contain sufficient information to permit a fairly accurate reconstruction of the source signal.

While the non-uniform sampling feature of asynchronous delta modulation points toward the possibility of efficient encoding, the continuous-valued time intervals between transmitted pulses do not offer compatibility with digital transmission or digital information storage.

In this paper we employ threshold sensing and digital coding of the time intervals between level crossings to mechanize a non-uniform sampling approach to source coding. Section 2 describes the non-uniform sampling encoding (NSE) approach. Section 3 discusses the incorporation of prediction to formulate a non-uniform sampling predictive coding (NSPC). Analytic and simulation results for a bandlimited Gaussian source are given in section 4.

2. NON-UNIFORM SAMPLING ENCODING

The motivation for the present proposed non-uniform sampling encoding scheme comes from the consideration that, if information about the source signal is carried in the times between level crossings rather than in the quantized amplitude as in DPCM systems, the encoder possesses an ability to follow the signal waveform even on a non-adaptive basis. The non-uniform sampling problem is depicted in Fig. 1 where the level crossing detector (LCD), a device consisting of a number of uniformly spaced levels, is used to monitor the level crossings. A signal sample is taken whenever a level crossing takes place. If a pulse is generated and transmitted everytime a level crossing occurs, the encoding procedure is called asynchronous delta modulation [8]. However, the time between level crossings is a continuous variable, which is not suitable for digital transmission or digital storage. The present study considers quantizing the time between level crossings and encoding the quantized time intervals in uniform steps. As a result, while the sampling is non-uniform, the information transmission is synchronous.

Let t_j be the time instant at which the source signal crosses a level and $\tau_j = t_j - t_{j-1}$ be the interval between the last level crossing at time t_{i-1} and the present crossing at time t_j. Let t_q be the time quantum so that the quantized time interval is prescribed by

$$\hat{\tau}_j = \begin{cases} t_q & \text{if } \tau_j \le t_q \\ m_j t_q & \text{if } t_q < \tau_j \end{cases} \tag{1}$$

where

$$m_j \triangleq \left\lceil \tau_j/t_q + 1/2 \right\rceil \qquad (2)$$

and the symbol $\lceil \cdot \rceil$ denotes the integer smaller than or equal to the argument. The error resulting from this time quantization is bounded by $\pm\, t_q/2$. The signal distortions due to time quantization for a bandlimited Gaussian signal is given in section 4.

The information about the source signal is contained in τ_j and the direction of level crossing, i.e., an upward or a downward crossing. Thus, two symbols are needed to represent the level crossings and a third symbol to represent the quantized intervals $\hat{\tau}_j$. In the present study we make the following binary 2-tuple assignments:

00 ~ coding of $\hat{\tau}_j$
01 ~ upward crossing
10 ~ downward crossing

For example, if the signal crosses a level at time t_j in the upward direction and $\hat{\tau}_j = m_j\, t_q$, then (m_j-1) pairs of 0's are used to represent the length of $\hat{\tau}_j$ and the 2-tuple 01 is used to represent the upward crossing. The binary representation is then $\underbrace{00\text{¦}00\text{¦}\ldots\text{¦}00}_{(m_j-1)\text{pairs}}\text{¦}01$. It is seen that 2 binary digits are used to represent the quantum step t_q. Thus, the frequency with which 2 bits of information are used to encode the time intervals is $f_q = 1/t_q$. If $\bar{m}_j$ is large, the binary sequence generated by the non-uniform sampling encoder (NSE) has a high probability of 0's, which is particularly suitable for further distortionless, such as run-length, encoding.

For the purpose of illustrating the signal encoding and reproduction, suppose the signal crosses level x_i at time t_{j-1} and level $x_{i+1} = x_i + d$, where d is the amplitude stepsize of the LCD, at time $t_j = t_{j-1} + \tau_j$, with an upward motion. Then $\hat{\tau}_j = m_j\, t_q$. Assuming $m_j = 4$, the binary representation is then 00¦00¦00¦01, where the 3 pairs of binary 0's carry the information about the length of $\hat{\tau}_j$ and the 2-tuple 01 indicates the upward motion. Since $\hat{\tau}_j$ is only an approximation to τ_j, the reproduced sample has value equal to $x_i + d$, but occurs at time $(t_j \pm t_q/2)$, as illustrated in Fig. 2.

Let f_c be the cut-off frequency of the source signal and $f_s = k(2f_c)$, $k \geq 1$, be the sampling frequency under a uniform sampling scheme, such as DPCM. With an N bit quantizer the transmission rate of a DPCM coder is N bits/source sample, where the samples are taken at the frequency f_s. Let $f_q = nf_s$, where $0 < n$. The transmission rate of the non-uniform sampling encoder, preferred to the frequency f_s, is 2n bits/source sample. A comparison of performance between a DPCM and the NSE for a bandlimited

Gaussian source is given in section 4.

3. NON-UNIFORM SAMPLING PREDICTIVE CODING

The encoding procedure depicted in Fig. 1 is analogous to PCM coding in the sense that no prior correlation removal has been carried out. Unlike PCM in which the information is carried in the amplitude, the NSE carries the information in the times between level crossings. If the mean interval between level crossings is long the probability distribution of the NSE output binary sequence will be highly asymmetric with P (0) >> P (1). The distortionless encoding stage of an overall data compression system can offer a high compression ratio. Therefore, for sources, such as video images, which exhibit short fast changing intervals separated by long relatively quiescent intervals, the tandem connection of the NSE and an entropy coder can offer very good rate-distortion characteristics.

The average length of time between level crossings can be increased by a decrease in the slope of the source signal. This observation led us to propose the non-uniform sampling predictive coding (NSPC) scheme shown in Fig. 3. The NSE performs an analog-to-digital coding. The device $(\text{NSE})^{-1}$ performs a pseudo digital-to-analog conversion. With a knowledge of how a source signal is non-uniformly sampled it is possible to reconstruct the source from the non-uniform samples [13]. For the present we employ a simple technique of joining the reconstructed samples by straight lines and low-pass filtering the resultant signal to obtain $\hat{e}(t)$. The predictor is optimized by solving the minimization problem

$$\min_{\{g_i\}} E\{[e'(t)]^2\} \tag{3}$$

where

$$e'(t) = x'(t) - \hat{x}'(t) \tag{4}$$

The predictor takes the form of a tapped-delay line with tap separation T_s such that $1/T_s \geq 2 f_c$. That is

$$\hat{x}'(t) = \sum_{i=1}^{K} g_i(t)\, y(t-iT_s) \tag{5}$$

The first derivative of $E\{[e'(t)]^2\}$ with respect to g_i is given by

$$\frac{\partial E[e'(t)]^2}{\partial g_i} = -2 E\{e'(t)\, y(t-iT_s)\} \tag{6}$$

Approximating $e'(t)$ by $\hat{e}'(t)$ and $E\{e'(t)\, y(t-iT_s)\}$ by its unbiased estimate $e'(t)\, y(t-iT_s)$, we can compute the g_i's iterative by the following recursive formula [6]

$$g_i(t) = g_i(t-T_s) - \alpha\, \hat{e}'(t)\, y(t-iT_s) \tag{7}$$

where α, bounded by $-2 < \alpha < 0$, is a gradient constant.

4. ANALYSIS AND SIMULATION RESULTS

In this section we focus attention on the encoding of a bandlimited Gaussian source.

4.1 Approximate Analysis of Time Quantization Noise

Consider the level crossing problem depicted in Fig. 4. The time between level crossings at times t_{j-1} and t_j is $\tau_j = t_j - t_{j-1}$. Let $\hat{\tau}_j$ be the quantized approximation computed by (1). The error is then $\Delta\tau_j = \tau_j - \hat{\tau}_j$ with $|\Delta\tau_j| \leq t_q/2$. The reconstructed sample will be the point D instead of B. The amplitude error is $e_j = D - C$. If we approximate the segment between A and B by a straight line, then from geometry, we have

$$\begin{aligned} e_j &= \frac{d}{\tau_j} \, \Delta\tau_j \\ &= x'(t_j) \, \Delta\tau_j \end{aligned} \tag{8}$$

Assumptions

i) $x'(t)$ and $\Delta\tau_j$ are statistically independent, and

ii) $\{\Delta\tau_j\}$ is zero mean and uniformly distributed in the interval $(-t_q/2, t_q/2)$.

With the above assumptions, we deduce that

$$E[e_j^2] = E\{[x'(t)]^2\} \cdot E\{[\Delta\tau_j]^2\}$$

By assumption ii) we have

$$\begin{aligned} E\{[\Delta\tau_j]^2\} &= \int_{-t_q/2}^{t_q/2} (\Delta\tau_j)^2 \cdot P(\Delta\tau_j) \, d(\Delta\tau_j) \\ &= t_q^2/12 \end{aligned} \tag{9}$$

On the assumption that $x'(t)$ has zero mean, then $\sigma_{x'}^2 = E\{[x'(t)]^2\}$. It follows that

$$\sigma_e^2 = E[e_j^2] = \sigma_{x'}^2 \cdot t_q^2/12 \tag{10}$$

Consider a bandlimited Gaussian source with spectral density

$$S(f) = \begin{cases} A & |f| \leq f_c \\ 0 & \text{otherwise} \end{cases}$$

Since $\sigma_x^2 = \int_{-f_c}^{f_c} S(f) \, df$, we have $A = \sigma_x^2/(2f_c)$. The variance of the slope is then

$$\begin{aligned} \sigma_{x'}^2 &= \int_{-f_c}^{f_c} (2\pi f)^2 \; S(f) \, df \\ &= (2\pi f_c)^2 \; \sigma_x^2/3 \end{aligned} \tag{11}$$

Substituting (11) in (10) we have

$$\sigma_e^2 = [\pi f_c/(3f_q)]^2 \; \sigma_x^2$$

where $f_q = 1/t_q$. The normalized distortion is thus given by

$$D \triangleq \sigma_e^2/\sigma_x^2 = [\pi f_c/(3f_q)]^2$$

$$= 0.40 + 20 \log_{10} (f_c/f_q) \quad \text{dB} \tag{12}$$

The transmission rate is 2 bits per f_q Hz of frequency. A plot of D as a function of (f_c/f_q) is shown in Fig. 5.

4.2 Mean Time Between Level Crossings

Following an argument similar to Blake and Lindsay [14], we use a first order approximation depicted in Fig. 6, where we examine the process in the proximity of $X = X_o$. The expected number of level crossings per unit time is

$$\nu(X_o) = \frac{\Pr[X_o < X < X_o + dx, \ -\infty < X' < \;]}{-dt}$$

$$= -\int_{-\infty} P(X_o, X') \left(\frac{dx}{dt}\right) dx'$$

The total frequency of crossings of this level is

$$V(X_o) = \int_{-\infty}^{\infty} |X'| \ P(X_o, X')dx'$$

For a Gaussian source the slope and the amplitude at a given instant are statistically independent. It can then be easily shown that

$$V(X_o) = \frac{2f_c}{\sqrt{3}} \quad \exp[- \frac{X_o^2}{2\sigma_x^2}]$$

With $X = n \cdot d$, $n = 0, \pm 1, \pm 2, \ldots$, the total frequency of crossings, denoted by r, is

$$r = V(0) + 2\sum_{n=1}^{\infty} V(nd)$$

$$= \frac{2f_c}{\sqrt{3}} + 2\left(\frac{2f_c}{\sqrt{3}}\right) \sum_{n=1}^{\infty} \exp[-\frac{1}{2} \frac{(nd)^2}{\sigma_x^2}] \tag{13}$$

where we have used the property $V(-X_o) = V(X_o)$ for a Gaussian process. For d in the range

$$0.01 \leq \frac{d}{\sigma_x} \leq 0.50$$

the frequency of level crossings can be approximated by

$$r \simeq 2 \sqrt{\frac{2\pi}{3}} \ (f_c \sigma_x/d) \tag{14}$$

Hence, the mean time between crossings is

$$\bar{\tau} = 1/r \simeq \frac{1}{2} \; \frac{3}{2\pi} \; \frac{1}{f_c} \; \frac{d}{\sigma_x} \tag{15}$$

4.3 Selection of Time Quantum t_q

The maximum slope, s_{max}, that can be accommodated by the NSE without experiencing overload is

$$s_{max} = d/t_q = d \cdot f_q \tag{16}$$

For the encoder to have an overload probability of less than 4×10^{-5}, we must have [9]

$$4\sigma_{x'} < s_{max} \tag{17}$$

For the bandlimited Gaussian source, we have previously shown that

$$\sigma_{x'}^2 = \frac{4}{3} \pi^2 f_c^2 \sigma_x^2 \tag{18}$$

Using (16) and (18) in (17), we obtain the inequality

$$\frac{8\pi}{\sqrt{3}} \; \frac{\sigma_x}{d} < f_q/f_c \tag{19}$$

from which we have

$$t_q < \frac{d}{\sigma_x} \cdot \frac{1}{f_c} \cdot \frac{\sqrt{3}}{8\pi} \tag{20}$$

4.4 Numerical Results

We have carried out a simulation study of the encoding of a bandlimited Gaussian source, which was generated using a 64th order transversal filter to give the resultant bandlimited Gaussian source the following spectral characteristics: i) passband ripple = ± 0.136%, ii) minimum stopband attennation = –57.4 dB, iii) $f_c = 0.12$ at the 3 dB point, and iv) $\sigma_x = 1$. With $d = 0.5$, we require $t_q < 0.2872$ or $f_q/f_c > 29.02$. With $t_q = 0.25$ we we encoded the bandlimited Gaussian source using NSE and NSPC. A segment of the bandlimited Gaussian process and the associated errors are shown in Fig. 7. It is observed that the errors resulting from the NSE and NSPC are comparable; the difference lies in that $\bar{\tau}_{NSPC} > \bar{\tau}_{NSE}$ so that the coded binary sequence at the output of the NSPC has a higher P(0).

For the bandlimited Gaussian source, we previously obtained

$$\bar{\tau}_{NSE} = \frac{1}{2} \sqrt{\frac{3}{2\pi}} \; \frac{1}{f_c} \; \frac{d}{\sigma_x} \tag{15}$$

With the above values of d, σ_x and f_c, we have $\bar{\tau}_{NSE} = 1.44$. It follows that $\bar{m} = \bar{\tau}_{NSE}/t_q = 5.8$ and P(0) = 0.91. From simulation we have P(0) = 0.92.

In the simulation we compute the signal-to-noise ratio by the following formula:

$$(S/N) = \frac{\sum_{j=1}^{J} x_j^2}{\sum_{j=1}^{J} (x_j - x_j^*)^2} \tag{21}$$

The simulation results for the NSE and NSPC encoders are plotted in Fig. 8 as a function of $F_s = f_q/f_c$. Also plotted is the result of an optimized linear delta modulator (LDM), i.e., the stepsize is optimized for each value of $F_s = f_s/f_c$. It is observed that the non-uniform sampling approach offers superior performance to the optimized LDM.

5 Concluding Remarks

We have described a non-uniform sampling approach to source coding, either with or without prediction. The information about the source signal is contained in the times between level crossings, which are coded into a highly asymmetric {0,1} binary sequence with P(0) >>P(1). For the band-limited Gaussian source considered, we have P(0) > 0.9. Simple run-length encoding [6] will provide an additional compression ratio greater than 2. This salient feature does not prevail in delta modulation systems.

It is expected that encoding of images with NSE or NSPC will yield a binary sequence with a much higher value of P(0) than the bandlimited Gaussian source. The resulting gain in further distortionless encoding can also be expected to be larger. Application to image coding is currently under investigation.

REFERENCES

1. J.B. O'Neal, Jr., "Predictive quantizing systems (differential pulse-code modulation) for the transmission of television signals," B.S.T.J., vol. 45. pp. 689-721, May-June 1966.
2. N.S. Jayant, "Digital coding of speech waveforms: PCM, DPCM, and DM quantizers," Proc. IEEE, vol. 62, pp. 611-632, May 1974.
3. J.E. Abate, "Linear and adaptive delta modulation," Proc. IEEE, vol. 55, pp. 298-308, March 1967.
4. N.S. Jayant, "Adaptive delta modulation with a one-bit memory," B.S.T.J. vol. 49, pp. 321-342, March 1970.
5. L.H. Zetterberg and J. Uddenfeldt, "Adaptive delta modulation with delayed decision," IEEE Trans. Communications, vol. COM-22, No. 9, pp. 1195-1198, Sept. 1974.
6. J.W. Mark, "Adaptive predictive run-length encoding for analogue sources," Proc. IEE, vol. 123, No. 11, pp. 1189-1196, Nov. 1976.
7. T.A. Hawkes and P.A. Simonpieri, "Signal coding using asynchronous delta modulation," IEEE Trans. Commun., vol.COM-22,No.3,pp.346-348,Mar.1974.
8. H. Inose, T. Aoki and K. Watanabe, "Asynchronous delta modulation systems," Electronics and Communications in Japan, pp.34-42, Mar. 1966.
9. R. Steele, Delta Modulation Systems, J.W. Arrowsmith Ltd. Bristol, England, 1975

10) E.N. Protonotario, "Slope overload noise in differential pulse code modulation," IEEE Trans. Info. Theory,vol.IT-20,No.1,pp.58-64,Jan.1974.

11) L.H. Goldstein and B. Liu, "Quantization noise in ADPCM systems," IEEE Trans. Communications, vol. COM-25, No. 2, pp. 227-238, Feb. 1977.

12) J. Max, "Quantizing for minimum distortion, "IRE Trans. Inform. Theory, vol. IT-6, pp. 7-12, March 1960.

13) J.L. Yen, "On nonuniform sampling of bandwidth-limited signals," IRE Trans. on Circuit Theory, pp. 251-257, Dec. 1956.

14) I.F. Blake and W.C. Linday, "Level-crossing problems for random processes," IEEE Trans. Inform. Theory, vol.IT-19,No.3.pp.295-315,May73.

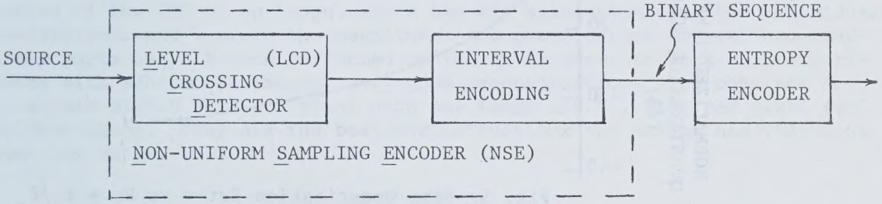

SOURCE
LEVEL (LCD) CROSSING DETECTOR
INTERVAL ENCODING
BINARY SEQUENCE
ENTROPY ENCODER
NON-UNIFORM SAMPLING ENCODER (NSE)

Fig. 1 Non-uniform Sampling Encoding

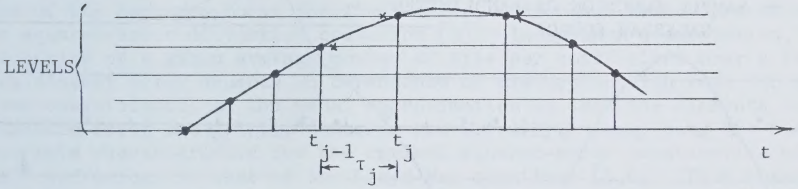

LEVELS
t_{j-1}
t_j
τ_j
t

Fig. 2 Illustration of Sample Reconstruction

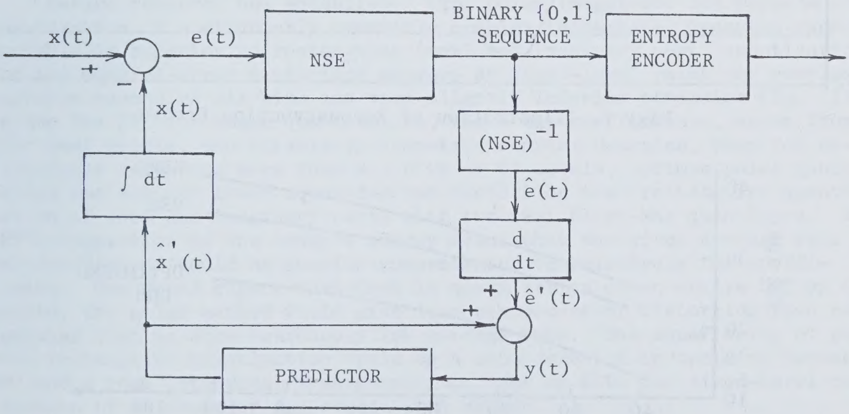

x(t)
+
−
e(t)
NSE
BINARY {0,1} SEQUENCE
ENTROPY ENCODER
x(t)
$(NSE)^{-1}$
$\hat{e}(t)$
$\int dt$
$\hat{x}'(t)$
$\frac{d}{dt}$
+
$\hat{e}'(t)$
+
PREDICTOR
y(t)

Fig. 3 Non-uniform Sampling Prdeictive Coding (NSPC)

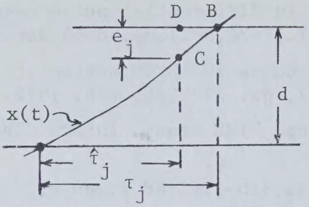

D
B
e_j
C
d
x(t)
τ_j
τ_j

Fig. 4 Time Quantization Error

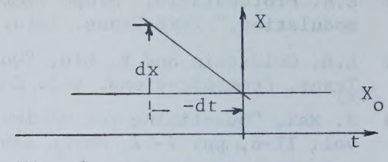


Fig. 6 Crossing at Level X_o

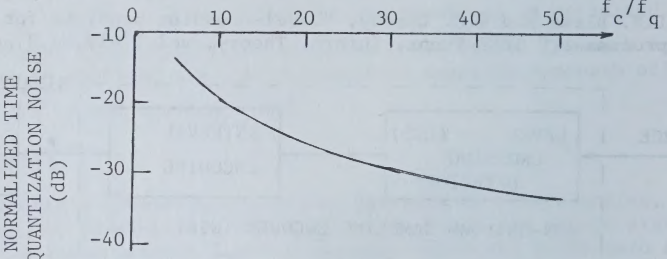


Fig. 5 Time Quantization Error vs $F_s = f_c/f_q$

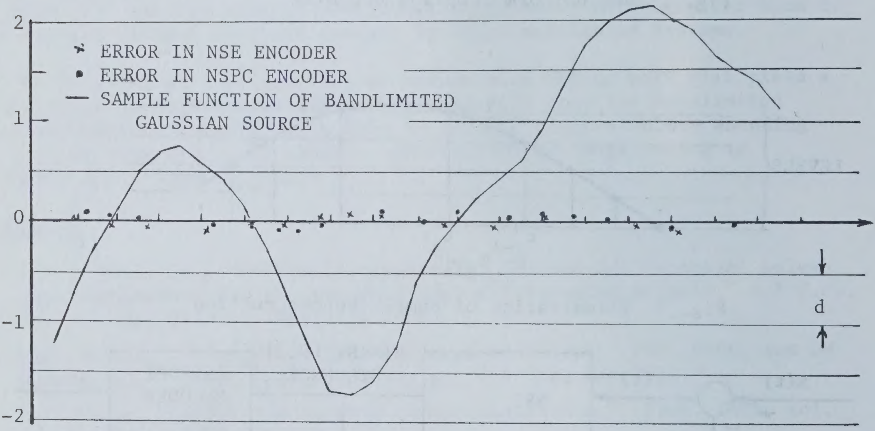


Fig. 7 Illustration of Reconstruction Errors

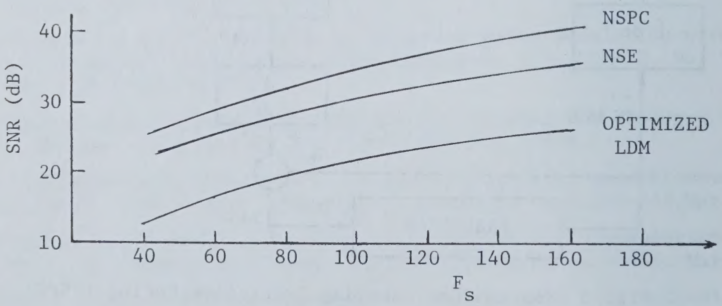


Fig. 8 Signal-to-Noise Ratio Performance of different Encoders

OPTIMUM FIXED-LEVEL QUANTIZATION OF THE DFT OF ACHROMATIC IMAGES

WILLIAM A. PEARLMAN
Electrical and Systems Engineering Department
Rensselaer Polytechnic Institute
Troy, New York 12181

ABSTRACT

Recent work has established the superiority of polar quantization over rectangular quantization for a circularly symmetric complex (bivariate) Gaussian random variable for a given number of levels. Using the concept of "useful" quantization levels, this work is extended to the optimal quantization of the DFT of an image, where the bit allocation over the individual coefficients and between the magnitudes and phase angles or real and imaginary parts has not been performed optimally in previous work. In experiments with achromatic images, very good reconstructions were obtained down to a rate of 0.5 bits per pixel with one image and 0.1 bits per pixel for another image. They are the best reconstructions the author has viewed at such low rates.

INTRODUCTION

There has been much progress in image coding through quantization of image transforms during the last decade. Recent efforts have focused on the use of the Discrete Cosine Transformation (DCT) (e.g., see [1,2]), as it appears to be a good approximation to the optimum decorrelating transformation of the Karhunen-Loeve for first-order, wide-sense Markov processes for the squared-error distortion criterion [3]. In optimum quantization, the allocation of a given average number of bits per coefficient over a transform element array depends on dependence of distortion with rate for any given coefficient. As the usual approximation is that the elements in the transform array are jointly Gaussian and mutually independent, the distortion-rate characteristic for the optimal squared-error quantization of any real coefficient is that of the Lloyd-Max quantizer [5,6]. This quantizer would operate on all elements of the DCT, because they are real.

Recent research has established that polar (magnitude and phase angle) quantization of a circularly symmetric complex (bivariate) Gaussian random-variable is superior to rectangular (real and imaginary part) quantization for the squared-error distortion measure at fixed-level rates per complex variable exceeding six bits and very slightly inferior otherwise [7]. If we use the DFT for image quantization, where the coefficients, aside from four real points, are circularly symmetric complex Gaussian, then for coefficients receiving more than six bits or 64 levels, optimum polar quantization can achieve lower squared-error distortion than rectangular quantization of real and imaginary parts with two real Lloyd-Max quantizers. The DFT's compaction of the image's energy means that the given average rate per coefficient would be heavily concentrated in relatively few coefficients. One would expect then that in quantization of an entire DFT by one method, the polar method would give less squared-error distortion than rectangular down to some reasonably low average rate. The superiority of polar over rectangular quantization would be a consideration in deciding between DFT and a real orthogonal transformation, such as DCT, for fixed-level quantization of an image or any source with memory.

Our purpose here is to perform minimum mean-squared error polar and rectangular DFT quantizations of achromatic images and compare the results at the same rate through subjective viewing and calculations of squared-error distortions. In order to accomplish the quantization of the DFT, one must first allocate the average rate per pixel optimally over the array of complex coefficients. (There are four real points which are treated appropriately). Then, secondly, the rate per complex coefficient must be divided optimally between real and imaginary parts or magnitude and phase angle. Actually, these two steps have not been performed optimally in previous image coding work for polar or rectangular DFT quantization. The optimal allocation of the given rate per pixel over the DFT array depends on the functional relationship between distortion and rate for each complex element [4]. When we allow only useful levels or rates, as defined by Pearlman [7], a different functional relationship is operative by which the distortion decrease is most rapid for increasing rate. Using the appropriate functions for polar and rectangular quantization, the optimum allocations of the rate over the DFT array can be readily determined by extending Segall's formulas [4] to our case. The usual procedure has been to allocate the rate according to the prescription of the rate distortion function. This procedure is non-optimal for our case. In the second step, to our knowledge, optimal polar quantization of individual coefficients has never been attempted. The reason is, perhaps, that its potential superiority over rectangular quantization has just recently been proved and that the independence and identical distribution of real and imaginary parts makes rectangular quantization easier to accomplish.

The image quantization experiments bear out the claims of lower squared error for polar quantization, but subjectively the images at the same rate were not discernibly different. The optimum quantization of the DFT, however, produced the best reconstructions the author has yet viewed at rates as low as 0.3 and 0.1 bits per pixel.

Optimal Quantization of the DFT

The problem is to quantize the two dimensional image DFT with minimum mean squared error at a given average rate per coefficient. We first approximate the DFT as a complex Gaussian process whose nonredundant coefficients are mutually independent. (Each redundant coefficient is one member of a complex conjugate pair arising from the transformation upon a real intensity array). The approximation is justified because of the asymptotic normality and independence of the nonredundant DFT coefficients [8] and the large number of 65,536 (256x256) picture elements (pixels or pels). The steps in the optimal quantization of the DFT process are the assignment of the proper number of bits and the specification of the decision intervals and quantization values for each coefficient. Here we chose optimal nonuniform quantization for the individual coefficients. The bit assignments for polar and rectangular quantizations of the complex elements will be somewhat different, because of their different distortion-rate characteristics.

The bit assignment to the coefficients of the real Karhunen-Loeve transformation on a real Gaussian process has been solved by Segall [6] in terms of the distortion-rate characteristics of coefficients' quantizers. Necessary conditions for his solution are that the distortion-rate characteristics be convex upward and have continuous first derivatives. For our purposes here, we note his solution to be valid for complex coefficients, but we must consider a case where the quantizers' distortion-rate characteristics are only piecewise convex and have isolated points with discontinuous first derivatives.

To simplify notation, let us consider the two-dimensional NxN DFT ordered into a one-dimensional array with N^2 elements. Let B_j denote the number of bits given to the j^{th} coefficient and $k_j(B_j)$ the distortion of the minimum mean-squared error quantizer of the j^{th} coefficient when scaled to unit variance and assigned B_j bits. The distortion versus rate function k_j is indexed since the real and complex coefficients have different quantizer characteristics. (There are four real coefficients in the two-dimensional DFT). We wish to find the allocation of the average number of bits per pixel R that minimizes

$$D = \sum_{j=1}^{N^2} \sigma_j^{\,2}\, k_j(B_j) \tag{1a}$$

subject to

$$\sum_{j=1}^{N^2} B_j = N^2 R \tag{1b}$$

and

$$B_j \geq 0 \quad \text{for} \quad j = 1, 2, \ldots, N^2 \tag{1c}$$

Following Segall [4], we can reformulate the problem in (1) as a minimization of the conjugate functional according to

$$\min_{\lambda} \{\lambda N^2 R - \sum_{j=1}^{N^2} \min_{B_j \geq 0} [B_j \lambda + \sigma_j^{\,2}\, k_j(B_j)]\} \tag{2}$$

We require then a series of one-dimensional minimizations prescribed by

$$\min_{B_j \geq 0} [B_j \lambda + \sigma_j^{\,2}\, k_j(B_j)].$$

The distortion-rate characteristics are empirically determined from piecewise linear least squares approximations of the logarithm of the distortion and hence have the form:

$$k(B) = \begin{cases} 2^{-\alpha_1 B}, & 0 \leq B < B_1 \\ a2^{-\alpha_2 B}, & B_1 \leq B < B_2 \\ b2^{-\alpha_3 B}, & B_2 \leq B \end{cases} \tag{3}$$

where $\alpha_1 < \alpha_2 < \alpha_3$ and the constants a and b are chosen so that k(B) is continuous. Note that k(B) is monotone decreasing with increasing B. Hence,

$$a = 2^{(\alpha_2 - \alpha_1)B_1}$$

$$b = 2^{[(\alpha_2 - \alpha_1)B_1 \; + \; (\alpha_3 - \alpha_2)B_2]}$$

The minimization is of the form:

$$\min_{B \geq 0} [B\lambda' + k(B)]$$

where $\lambda' = \lambda/\sigma^2$, omitting subscripts for convenience. If k(B) were convex with a continuous first derivative, k'(B), the bit assignment would be

$$B = 0 \quad \text{for} \quad \lambda' \geq -\,k'(0)$$

and at the solution of

$$\lambda' + k'(B) = 0 \quad \text{for} \quad 0 < \lambda' \leq -\,k'(0).$$

But for k(B) considered here the solution of the above equation would not yield unique values of B for certain values of λ', since there are disjoint regions of B where k'(B) is the same. There is, however, a unique minimum to $\lambda' B + k(B)$ since k(B) is monotone decreasing and $\lambda' B$ is monotone increasing for a given λ'. Solving for the minimum for every λ', we obtain

$$B = \begin{cases} 0, \quad \lambda' \geq -\,k'(0) \\ h(-\lambda'), \ -k'(0) > \lambda' \geq -\,k'(B_{1c}), \ 0 < B \leq B_{1c} < B_1 \\ h(-\lambda'), \ -k'(B_{1c}) > \lambda' \geq -\,k'(B_{2c}), \ B_1 \leq B \leq B_{2c} < B_2 \\ h(-\lambda'), \ -k'(B_{2c}) > \lambda' > 0, \ B \geq B_2 \end{cases} \tag{4}$$

where h is the inverse function of k' in the given range, B_{1c} is the point between B=0 and $B=B_1$ such that

$$-k'(B_{1c}) = \frac{k(B_1 - \frac{1}{\alpha_1}\ \log_2 \frac{\alpha_2}{\alpha_1}) - k(B_1)}{\frac{1}{\alpha_1}\ \log_2 \frac{\alpha_2}{\alpha_1}} \tag{5a}$$

and B_{2c} is the point between $B=B_1$ and $B=B_2$ such that

$$-k'(B_{2c}) = \frac{k(B_2 - \frac{1}{\alpha_2}\ \log_2 \frac{\alpha_3}{\alpha_2}) - k(B_2)}{\frac{1}{\alpha_2}\ \log_2 \frac{\alpha_3}{\alpha_2}} \tag{5b}$$

There are three different functions k(B) of the given form corresponding to the two types of complex variable quantizations and one type of real variable quantization. The functions k(B) for complex variables, the parameters of which are given below, were obtained by retaining only the useful quantization level numbers. These level numbers are the ones whose factors are closest to the ideal for magnitude and phase angle or real and imaginary parts and produce a rapid and smoothly decreasing values of k(B) for increasing B. We state the values of the three sets of parameters for k(B) and the corresponding bit assignment solutions below. The solutions are expressed in terms of another parameter Θ related to λ and λ' by $\Theta = -\lambda/k'(0) = -\sigma^2\lambda'/k'(0)$.

(i) Complex Polar Quantization:

$B_1 = 3.3219$, $B_1 = 5.9069$, $\alpha_1 = 0.7293$, $\alpha_2 = 0.9105$, $\alpha_3 = 0.9701$

$$B = \begin{cases} 0, \ \Theta \geq \sigma^2 \\ \frac{1}{0.7293}\ \log_2\ (\frac{\sigma^2}{\Theta})\ , \ \sigma^2 > \Theta \geq 0.1999\sigma^2 \\ \frac{1}{0.9105}\ \log_2\ (\frac{1.9196\sigma^2}{\Theta})\ , \ 0.1999\sigma^2 > \Theta \geq 0.0476\sigma^2 \\ \frac{1}{0.9701}\ \log_2\ (\frac{2.6099\sigma^2}{\Theta})\ , \ 0.04763\sigma^2 > \Theta > 0 \end{cases} \tag{6}$$

(ii) Complex Rectangular Quantization:

$$B_1 = 2.5850,\ B_2 = 5.6147,\ \alpha_1 = 0.7169,\ \alpha_2 = 0.8628,\ \alpha_3 = 0.9480$$

$$B = \begin{cases} 0, \ \Theta \geq \sigma^2 \\ \dfrac{1}{0.7169} \ \log_2 \left(\dfrac{\sigma^2}{\Theta}\right), \ \sigma^2 > \Theta \geq 0.3041\sigma^2 \\ \dfrac{1}{0.8626} \ \log_2 \left(\dfrac{1.5632\sigma^2}{\Theta}\right), \ 0.3041\sigma^2 > \Theta \geq 0.5703\sigma^2 \\ \dfrac{1}{0.9480} \ \log_2 \left(\dfrac{2.3926\sigma^2}{\Theta}\right), \ 0.05703\sigma^2 > \Theta > 0 \end{cases} \tag{7}$$

(iii) Optimum Real Quantization (using Segall's formulas [4]):

$$B_1 = 2.32,\ \alpha_1 = 1.57,\ \alpha_2 = 1.78 \quad (\alpha_3 = \alpha_2,\ B_2 \text{ nonexistent})$$

$$B = \begin{cases} 0, \ \Theta \geq \sigma^2 \\ \dfrac{1}{1.57} \ \log_2 \left(\dfrac{\sigma^2}{\Theta}\right), \ \sigma^2 > \Theta \geq 0.083\sigma^2 \\ \dfrac{1}{1.78} \ \log_2 \left(\dfrac{1.46\sigma^2}{\Theta}\right), \ 0.083\sigma^2 > \Theta > 0 \end{cases} \tag{8}$$

The parameter Θ is found from the total fixed-rate constraint in (1b). The distortion versus rate formulas for k(B) for polar and rectangular quantization given in (i) and (ii) above are slightly different than those given previously by Pearlman [7] for useful levels in the highest rate regions, because piecewise linear least squares fits are made to the logarithm of the full curves for all rates up to ten bits and result in different starting points for the highest rate segments. At smaller rates, the logarithm of the distortion is not really linear with rate (see figures in [7]), so the fit with two straight lines is not an entirely accurate model at these rates. The average squared error is quite small however, so the accuracy is sufficient for our purposes here.

Once the number of bits is assigned to each complex DFT coefficient through the above formulas, we use Pearlman's tables [7] to round off the corresponding number of quantization levels (2^B) to the nearest useful level number and find the optimal number of levels for magnitude and phase angle for polar quantization or real and imaginary parts for rectangular quantization. (The numbers of levels for the real and imaginary parts are either equal or differ by one, whereas the average ratio of phase angle to magnitude levels is 2.89). Then the quantization of the value of a complex coefficient can be implemented by the specification of the decision intervals and quantization levels either for magnitude and phase angle given by Pearlman and Senge [9] or for real and imaginary parts given by Max [6].

Quantization of Images and Results

We wish now to enact optimal polar and rectangular quantizations for the DFT of actual images and assess the results both subjectively and in terms of average squared error. Two achromatic images were chosen for the quantization experiments: one a head-and-shoulders portrait of a woman and the other a moon landscape, both shown in Figure 1. The entire image of a woman has a high degree of contrast and detail, while the image of the moon landscape has relatively little contrast. The DFT variances are the coefficients in the power spectrum estimate, which is calculated from averaging the modified periodograms of the image's sub-blocks, analagous to the one-dimensional segmentation method of Welch [10]. Average rates of 2.0, 1.0, 0.5, 0.3, 0.1 and 0.05 bits per pixel were chosen for the quantizations of the images.

The normalized average squared errors of the reconstructions are tabulated in Table I. Normalized average squared error is defined as the average squared error divided by the image's average squared deviation from its mean. The polar DFT quantizations have lower error than the rectangular ones at all but the lowest two rates where they are in one case the same. (Actually, the polar error is smaller in all cases if one adds another decimal place in the error tabulation). If one compares the normalized error of the moon image with that of the corresponding woman image at the same rate and quantization method, it appears that the moon image's error is always larger. The unnormalized error is actually smaller in all cases, because the moon image's signal power is one-fifth that of the woman image.

Table 1

Calculations of Mean-Squared Error Versus Rate for Two Source Images
in Polar and Rectangular DFT Quantization

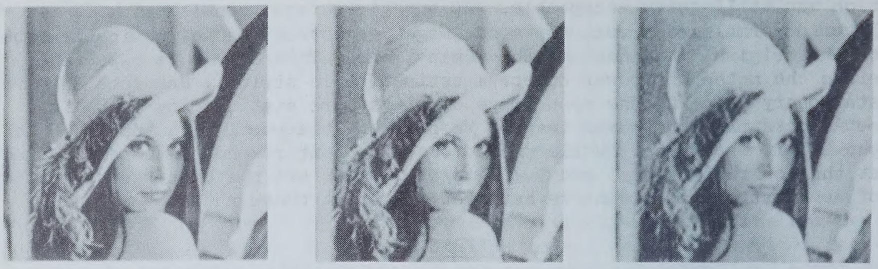
	Normalized MSE*	
Bits/ Pixel	Polar	Rectangular
	(a) Woman's Face	
2	.540E-02	.601E-02
1	.158E-01	.163E-01
0.5	.317E-01	.323E-01
0.3	.476E-01	.485E-01
0.1	.976E-01	.976E-01
0.05	.139	.141
	Signal Power = 2734	
	(b) Moon Landscape	
2	.114E-01	.121E-01
1	.335E-01	.341E-01
0.5	.628E-01	.632E-01
0.3	.898E-01	.907E-01
0.1	.161	.162
0.05	.220	.220
	Signal Power = 545	

*Normalized MSE = average squared error per pixel divided by the image's average squared deviation from its mean (signal power)

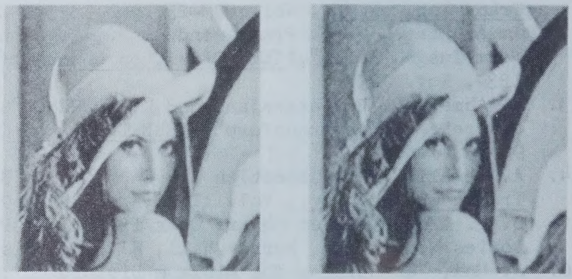
In a subjective comparison of polar and rectangular quantization there appeared to be little or no difference at all rates. The rates of 0.5 and 0.3 bits per pixel, shown in Figures 1 and 2, are illustrative of the results. One cannot discern any appreciable difference between the same images at the same rate quantized by the two methods. What is striking, however, is the high quality of the reconstructions at such low rates. In fact, the reconstructed images for 2.0 and 1.0 bits/pel (not shown) were practically the same as the originals. At 0.5 bits/pel the degradation just started to become noticable. Even at 0.1 bits/pel the moon image reconstruction was still quite acceptable. At any given rate, consistent with its lower unnormalized error, the moon image reconstruction was noticably truer to the original than that of the woman image. This result is expected, since the method of power spectrum estimation is strictly accurate only for stationary data and the moon image displays more stationarity among its sub-blocks than the woman image. We conclude that optimal quantization of the DFT is indeed worthwhile due to the excellent reconstructions obtained at the low rates of 0.5 and 0.3 bits/pel. They are the best reconstructions of achromatic images that we have yet viewed at these rates.

REFERENCES

1. T. R. Natarajan and N. Ahmed, "On Interframe Transform Coding", IEEE Trans. on Commun., Vol. COM-25, pp. 1323-1329, Nov. 1977.
2. J. A. Roese, W. K. Pratt, and G. S. Robinson, "Interframe Cosine Transform Image Coding", IEEE Trans. on Commun., Vol. COM-25, pp. 1329-1339, Nov. 1977.
3. N. Ahmed, T. R. Natarajan, and K. R. Rao, "On Image Processing and a Discrete Cosine Transform", IEEE Trans. on Computers, Vol. C-23, pp. 90-93, Jan. 1974.
4. A. Segall, "Bit Allocation and Encoding of Stationary Sources", IEEE Trans. Inform. Thy., Vol. IT-22, pp. 162-169, March 1976.
5. S. P. Lloyd, "Least Square Quantization in PCM", Bell Telephone Laboratories Memorandum, Murray Hill, NJ (unpublished).
6. J. Max, "Quantizing for Minimum Distortion", IEEE Trans. Inform. Thy., Vol. IT-6, pp. 7-12, March 1960.
7. W. A. Pearlman, "Polar Quantization of a Complex Gaussian Random Variable", IEEE Trans. Commun., Vol. COM-27, pp. 892-899, June 1979.
8. W. A. Pearlman and R. M. Gray, "Source Coding of the Discrete Fourier Transform", IEEE Trans. on Inform. Thy., Vol. IT-24, pp. 683-692, Nov. 1978.
9. W. A. Pearlman and G. H. Senge, "Optimal Quantization of the Rayleigh Probability Distribution", IEEE Trans. Commun., Vol. COM-27, pp. 101-112, Jan. 1979.
10. P. D. Welch, "The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging over Short, Modified Periodgrams", IEEE Trans. on Audio and Electro-acoustics, Vol. AU-15, pp. 70-73, June 1967.

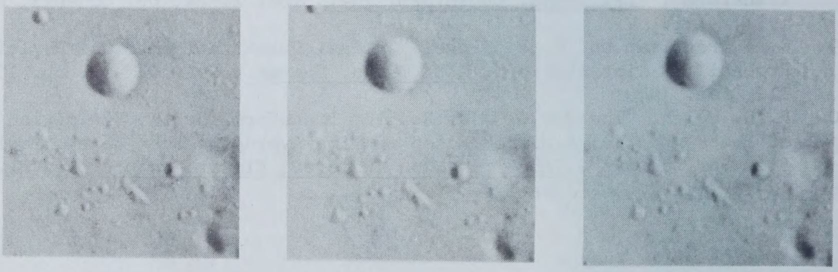


a) Original b) 0.5 bit rectangular c) 0.3 bit rectangular

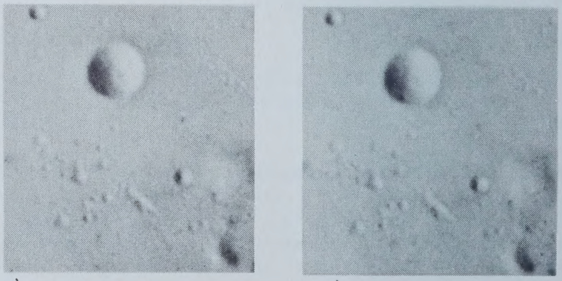


d) 0.5 bit polar e) 0.3 bit polar

Figure 1. Polar and Rectangular DFT Quantization of Woman Image at 0.5 and 0.3 bits/pel.



a) Original b) 0.5 bit rectangular c) 0.3 bit rectangular



d) 0.5 bit polar e) 0.3 bit polar

Figure 2. Polar and Rectangular DFT Quantization of Moon Landscape Image at 0.5 and 0.3 bits/pel.

SOME PROPERTIES OF UNIFORM STEP SIZE QUANTIZERS*

JAMES A. BUCKLEW and NEAL C. GALLAGHER, JR.
School of Electrical Engineering
Purdue University
West Lafayette, IN 47907

*This research was supported by The Air Force Office of Scientific Research under grant AFOSR 78-3605.

ABSTRACT

This paper treats some properties of the optimal mean square error uniform quantizer. It is shown for the OUQ that the mean square error (mse) is given by the input variance minus the output variance. It is shown that $\lim_{N\to\infty} \frac{mse}{\Delta^2/12} \geq 1$ with equality when the support of the random variable is contained in a finite interval. A class of probability densities which have the above limit greater than 1 is given. It is shown that $\lim_{N\to\infty} N^2\, mse = (b-a)^2/12$ where (b-a) is the measure of the smallest interval that contains the support of the input random variable.

In many problems arising in the evaluation or design of a control or communication system, it becomes necessary to predict the performance of a uniform quantizer. Uniform quantizers are of interest because they are usually the simplest quantizer structure to implement. The study of uniform quantization is also of interest because many noise processes in physical systems may be considered to be the noise produced by a uniform quantizing operation. For example the final position of a stepping motor or the line drawn by the pen of a computer plotting device under a continuous control may be considered to be corrupted by a uniform quantizing operation.

Because of the importance of these quantizers several authors have considered various properties of them. Widrow [1] shows that under certain conditions on the characteristic function of the input random variable, the quantization noise is uniformly distributed. Gish and Pierce [2] show that asymptotically the uniform quantizer is optimum in the sense of minimizing the output entropy subject to a fixed mean square error value. Sripad and Snyder [3] later extend Widrow's work to give a sufficient condition for when the quantization error is uniform and uncorrelated with the input random variable.

We will now state and prove some additional properties of these quantizers when we design them to minimize the mean square error. We may write down the analytic expression for the quantizer characteristic g(x) as,

$$g(x) = \begin{cases} a & x<q \\ a+(i+1)\Delta & q+i\Delta<x<q+(i+1)\Delta \quad i=0,\dots,N-3 \\ a+(N-1)\Delta & x>(N-2)\Delta+q \end{cases} \tag{1}$$

where N is the number of output levels in the quantizer. We see that if x is less than q or greater than q + (N-2)Δ, then x is truncated to a and a + (N-1)Δ respectively. An important parameter of interest is the width of the nontruncation region which equals (N-2)Δ.

The quantizer characteristic g(x) must be optimized with respect to three parameters, q which fixes its position along the x axis, a which fixes its position along the y axis, and Δ which specifies the step size of the quantizer. Because it makes little sense to speak of minimizing the mean square error of a random variable with infinite variance, we will always assume $\int_{-\infty}^{\infty} x^2 f(x)\, dx < \infty$.

Property 1

The optimum uniform quantizer preserves the mean of the input random variable.

Proof:

Suppose g(x) is the optimum uniform quantizer. Then we must have

$$\frac{\partial}{\partial \varepsilon} \int (x - g(x) + \varepsilon)^2 f(x)\, dx \,\Big|_{\varepsilon=0} = 0. \tag{2}$$

This implies,

$$\int x\, f(x) = \int g(x)\, f(x). \tag{3}$$

□

Property 2

For the optimum uniform quantizer we have that

$$a = q - \Delta/2.$$

Proof:

Suppose g(x) is the optimum uniform quantizer. Then we must have

$$\frac{\partial}{\partial \varepsilon} \int (g(x - \varepsilon) - x)^2 f(x)\, dx \,\Big|_{\varepsilon=0} = 0. \tag{4}$$

$$= \left[\frac{\partial}{\partial \varepsilon} \int g(x-\varepsilon)^2 f(x)\, dx - \frac{\partial}{\partial \varepsilon} \int 2xg(x-\varepsilon)\, f(x)\, dx\right] \Big|_{\varepsilon = 0}. \tag{5}$$

$$= \frac{\partial}{\partial \varepsilon}\left[\sum_{i=0}^{N-3} (a+(i+1)\Delta)^2 \int_{q+\varepsilon+i\Delta}^{q+\varepsilon+(i+1)\Delta} f(x)\, dx + a^2 \int_{-\infty}^{q+\varepsilon} f(x)\, dx\right.$$

$$+ (a+(N-1)\Delta)^2 \int_{q+\varepsilon+(N-2)\Delta}^{\infty} f(x)\, dx$$

$$- 2\left[\sum_{i=0}^{N-3} (a+(i+1)\Delta) \int_{q+\varepsilon+i\Delta}^{q+\varepsilon+(i+1)\Delta} xf(x)\, dx + a \int_{-\infty}^{q+\varepsilon} xf(x)\, dx\right.$$

$$+ (a+(N-1)\Delta) \int_{q+\varepsilon+(N-2)\Delta}^{\infty} xf(x)\, dx \qquad \Big|_{\varepsilon=0} = 0. \tag{6}$$

$$= \left[\sum_{i=0}^{N-3} (a+(i+1)\Delta)^2 \,[f(q+\varepsilon+(i+1)\Delta) - f(q+\varepsilon+i\Delta)]\right.$$

$$+ a^2 f(q+\varepsilon) - (a+(N-1)\Delta)^2 f(q+\varepsilon+(N-2)\Delta)$$

$$- 2\left[\sum_{i=0}^{N-3} (a+(i+1)\Delta)\ [(q+\varepsilon+(i+1)\Delta)\ f(q+\varepsilon+(i+1)\Delta)\right.$$

$$- (q+\varepsilon+i\Delta)\ f(q+\varepsilon+i\Delta)] + a(q+\varepsilon)\ f(q+\varepsilon)$$

$$- (a+(N-1)\Delta)\ (q+\varepsilon+(N-2)\Delta)\ f(q+\varepsilon+(N-2)\Delta) \qquad \Big|_{\varepsilon=0} = 0. \tag{7}$$

Simplifying this expression we obtain

$$(\Delta+2a-2q) \sum_{i=0}^{N-2} f(q+i\Delta) = 0.$$

The solution $\sum_{i=0}^{N-2} f(q+i\Delta) = 0$ is of no interest because without affecting the mean square error, we may always arbitrarily set $f(q+i\Delta) = 0, i = 0,\ldots,N-2$. Hence $\Delta+2a-2q = 0$ which is what we wish to prove.

[]

Property 3

The mean square error of an optimum uniform quantizer is given by the input variance minus the output variance.

Proof:

We again write the mean square error mse as

$$\text{mse} = E\{x^2\} - 2\ E\{xg(x)\} + E\{g(x)^2\}. \tag{8}$$

We wish to optimize this equation with respect to Δ. Using $a = q-\Delta/2$ we first obtain

$$E\{xg(x)\} = \sum_{i=0}^{N-3} (q+(i+\tfrac{1}{2})\Delta)^2 \int_{q+i\Delta}^{q+(i+1)\Delta} xf(x)\ dx$$

$$+ (q-\Delta/2) \int_{-\infty}^{q} xf(x)\ dx + (q+(N-\tfrac{3}{2})\Delta) \int_{q+(N-2)\Delta}^{\infty} xf(x)\ dx \tag{9}$$

and

$$E\{g(x)^2\} = \sum_{i=0}^{N-3} (q+(i+\tfrac{1}{2})\Delta)^2 \int_{q+i\Delta}^{q+(i+1)\Delta} f(x)\ dx$$

$$+ (q-\tfrac{\Delta}{2})^2 \int_{-\infty}^{q} f(x)\ dx + (q+(N-\tfrac{3}{2})\Delta)^2 \int_{q+(N-2)\Delta}^{\infty} f(x)\ dx. \tag{10}$$

Now substitute Eq. (9) and Eq. (10) into Eq. (8); take the partial derivative with respect to Δ and set the result equal to zero. We find that

$$E\{xg(x)\} + qE\{g(x)\} = E\{g(x)^2\} + qE\{x\}. \tag{11}$$

But $E\{g(x)\} = E\{x\}$ for the optimum quantizer. Hence $E\{xg(x)\} = E\{g(x)^2\}$ and we have for the mean square error mse

$$\text{mse} = E\{x^2\} - E\{g(x)^2\} \tag{12}$$

which together with Property 1 finishes the proof.

[]

Sripad and Snyder [3] show sufficient conditions for (x-g(x)) to be uniform and uncorrelated with x to be

$$\phi_x\left(\frac{2\pi n}{\Delta}\right) = \dot{\phi}_x\left(\frac{2\pi n}{\Delta}\right) = 0 \qquad n = \pm 1, \pm 2, \ldots \tag{13}$$

where $\phi_x(\omega)$ is the characteristic function of the input random variable x. Frequently in the analysis of a system corrupted by a uniform quantizing operation the assumption is made that the quantization noise is uncorrelated (sometimes independence is assumed) with the input. The next property demonstrates that this can't be done with the optimum uniform quantizer.

Property 4

Suppose the input probability density is Riemann integrable. Then the quantization noise can't be uncorrelated with the input for the optimum uniform quantizer.

Proof:

Without loss of generality assume E{X} = 0. Now suppose the converse to the property. This implies

$$E\{(x-g(x))x\} = E\{x^2\} - E\{g(x)x\} = 0 \tag{14}$$

But from Property 3

$$\begin{aligned} &E\{xg(x)\} = E\{g(x)^2\} \qquad \text{hence} \\ &E\{x^2\} - E\{g(x)^2\} = 0 \end{aligned} \tag{15}$$

But again from Property 3, the left hand side of Eq. (15) is the mean square error which implies a contradiction. That a probability density function is Riemann integrable necessarily implies that the mean square error for any finite number of output levels is greater than zero (i.e. f(x) has no delta functions).

[]

We now state an obvious property which will be used in several subsequent proofs. The proof of property 5 follows from a simple application of the Lebesgue dominated convergence theorem.

Property 5

The mean square error approaches zero for the optimal uniform quantizer as the number of output levels approaches ∞.

Let I = [a,b] be the smallest interval such that $\int_a^b f(x)\,dx = 1$. Note that |a| or |b| may be infinite.

Property 6

Suppose f(x) is Riemann integrable. Then for the optimum uniform quantizer, $\lim_{N\to\infty} (N-2)\Delta = b-a$.

Proof:

Suppose $\lim_{N\to\infty} (N-2)\Delta < b-a$. This implies for N sufficiently large that we are always truncating some finite amount of probability mass which means the mean square error can't go to zero which is a contradiction of the previous property. Hence we have the $\lim_{N\to\infty} (N-2)\Delta \geq b-a$.

Let us suppose $\lim_{N\to\infty} (N-2)\Delta > b-a$. Note that this makes sense only if the random variable is of finite support. Now for N large enough there is no truncation error. It is easy to show as will be done in the next property that for a quantizer with no truncation error, $\lim_{N\to\infty} \frac{mse}{\Delta^2/12} = 1$ for a Riemann integrable density function. So for N sufficiently large $(N-2)\Delta > C > b-a < \infty$. Then

$$1 = \lim_{N\to\infty} \frac{mse}{\Delta^2/12} \leq \lim_{N\to\infty} \frac{mse}{C^2/12(N-2)^2} \text{ or}$$

$$\lim_{N\to\infty} (N-2)^2 \, mse \geq \frac{C^2}{12} \tag{16}$$

Consider now a suboptimal quantizer whose input intervals are given by dividing up the interval I into N-2 equal subintervals. Denote the mean square error or this quantizer as mse_{SUB}, and its step size $\Delta_S = (b-a)/(N-2)$. This quantizer has no truncation error and hence

$$1 = \lim_{N\to\infty} \frac{mse_{SUB}}{\Delta_S^2/12} = \lim_{N\to\infty} \frac{mse_{SUB}}{(b-a)^2/12(N-2)^2} \text{ or}$$

$$\lim_{N\to\infty} (N-2)^2 \, mse_{SUB} = \frac{(b-a)^2}{12} < \frac{C^2}{12} \leq \lim_{N\to\infty} (N-2)^2 \, mse \tag{17}$$

which is a contradiction since we have found a suboptimal quantizer with a better mean square error than the optimal.

□

Property 7

Suppose the density function is Riemann integrable and $(b-a) < \infty$. Then for the optimal uniform quantizer we have

$$\lim_{N\to\infty} \frac{mse}{\Delta^2/12} = 1.$$

Proof:

From property 6 we know that $\lim_{N\to\infty} (N-2)\,\Delta_0 = b-a < \infty$ where Δ_0 is the optimum Δ. We may design a suboptimum quantizer by dividing the interval I (smallest interval such that $\int_a^b f(x)\, dx = 1$) into N-2 equal subintervals and use these subintervals as the breakpoints for our quantizer. We will denote the mean square error associated with this quantizer as mse_{SUB} and the step size $\Delta_S \overset{\Delta}{=} (b-a)/(N-2)$.

Define

$$M_i \overset{\Delta}{=} \underset{X\varepsilon(q+i\Delta)\ q+(i+1)\Delta)}{\mathrm{Sup} f(x)}$$

and

$$m_i = \underset{X\varepsilon(q+i\Delta)\ q+(i+1)\Delta)}{\infty\ f(x)} \quad .$$

Then since there is no truncation error for the suboptimal quantizer we have

$$\sum_{i=0}^{N-3} m_i \int_{q+i\Delta_S}^{q+(i+1)\Delta_S} (X-(q+(i+\tfrac{1}{2})\Delta_S))^2\, dx \leq mse_{SUB}$$

and

$$mse_{SUB} \leq \sum_{i=0}^{N-3} M_i \int_{q+i\Delta_S}^{q+(i+1)\Delta_S} (X-(q+(i+\tfrac{1}{2})\Delta_S))^2\, dx \tag{18}$$

or

$$\frac{\Delta_S^2}{12} \sum_{i=0}^{N-3} m_i \Delta_S \leq mse_{SUB} \leq \frac{\Delta_S^2}{12} \sum_{i=0}^{N-3} M_i \Delta_S \tag{19}$$

$$\lim_{N\to\infty} \sum_{i=0}^{N-3} m_i \Delta_S \leq \lim_{N\to\infty} \frac{mse_{SUB}}{\Delta_S^2/12} \leq \lim_{N\to\infty} \sum_{i=0}^{N-3} M_i \Delta_S \tag{20}$$

Now since f(x) is a density function and is Riemann integrable

$$\lim_{N\to\infty} \sum_{i=0}^{N-3} m_i \Delta_S = \lim_{N\to\infty} \sum_{i=0}^{N-3} M_i \Delta_S = 1$$

implies

$$\lim_{N\to\infty} \frac{mse_{SUB}}{\Delta_S^2/12} = 1. \tag{21}$$

Now $\lim_{N\to\infty} \frac{\Delta_S}{\Delta_0} = \lim_{N\to\infty} \frac{(N-2)\Delta_S}{(N-2)\Delta_0} = \frac{\lim_{N\to\infty}(N-2)\Delta_S}{\lim_{N\to\infty}(N-2)\Delta_0} = 1,$ which gives automatically $\lim_{N\to\infty} \frac{\Delta_S^2}{\Delta_0^2} = 1.$ Now for any quantizer whose nontruncation region covers the support of the Riemann integrable density function in the limit as N approaches infinity, we may show as above that $\lim_{N\to\infty} \frac{mse}{\Delta^2/12} \geq 1$. This bound is arrived at by ignoring the truncation error and is true for finite or infinite support density functions. We now have the following string,

$$\lim_{N\to\infty} \frac{mse_{SUB}}{\Delta_0^2/12} = \lim_{N\to\infty} \left(\frac{mse_{SUB}}{\Delta_S^2/12}\right) \left(\frac{\Delta_S^2/12}{\Delta_0^2/12}\right)$$

$$= \left(\lim_{N\to\infty} \frac{mse_{SUB}}{\Delta_S^2/12}\right) \left(\lim_{N\to\infty} \frac{\Delta_S^2/12}{\Delta_0^2/12}\right) = 1 \qquad (22)$$

but

$$1 = \lim_{N\to\infty} \frac{mse_{SUB}}{\Delta_0^2/12} \geq \lim_{N\to\infty} \frac{mse_{OPTIMAL}}{\Delta_0^2/12} \geq 1 \qquad (23)$$

$$\text{Or } \lim_{N\to\infty} \frac{mse_{OPTIMAL}}{\Delta_0^2/12} = 1$$

which is what we wanted to prove.

[]

Zador [4] shows that if f(x) is Riemann integrable and $E\{x^{2+\delta}\} < \infty$ for same $\delta > 0$; then we have for the optimal nonuniform quantizer

$$\lim_{N\to\infty} N^2 \, mse = ||f||_{1/3}/12$$

where $||f||_{1/3}$ is the $L_{1/3}$ norm. This result shows that for the nonuniform quantizer, the mean square error decreases on the order of $1/N^2$ for large N. Is there a similar property for the optimum uniform quantizer? We now give our next property.

Property 8

Suppose f(x) is Riemann integrable. Then for the optimum uniform quantizer $\lim_{N\to\infty} N^2 \, mse = \frac{(b-a)^2}{12}$.

Proof:

Suppose $(b-a) = \infty$. Then $1 \leq \lim_{N\to\infty} \frac{mse}{\Delta^2/12} = \lim_{N\to\infty} \frac{(N-2)^2 mse}{(N-2)^2 \Delta^2/12}$

$$= \frac{\lim (N-2)^2 mse}{\lim N^2 \Delta^2/12} \qquad (24)$$

but $(N-2)^2 \, \Delta^2 \to \infty$ which implies $\lim_{N\to\infty} (N-2)^2 \; mse \to \infty$.

If $b-a < \infty$ then $\lim_{N\to\infty} \frac{mse}{\Delta^2/12} = 1$ or $\lim_{N\to\infty}(N-2)^2 mse = \lim_{N\to\infty} N^2 mse =$

$$\frac{\lim (N-2)^2 \Delta^2}{12} = \frac{(b-a)^2}{12}$$ which finishes the proof.

Discussion

We should note that not everyone employs the same definition of optimum uniform quantizer that we have used. For example Pearlman and Senge [5] have published tables of the optimal uniform Rayleigh quantizer. For their computations, they add the constraints $a = 0$ and that $q = \Delta/2$.

It is interesting to note that properties 1 and 3 are also shared by the optimal nonuniform quantizer as shown in [6]. As a further consequence of these two properties we find that for the $N=2$ case, the optimum uniform quantizer and the optimum nonuniform quantizer are identical.

Property 7 is one of the more interesting properties proved in this paper. A common approximation to the mean square error of a uniform quantizer has been $\Delta^2/12$. Consider the class of density functions given by

$$f(x) = \frac{(1 + \frac{\delta}{2})}{(1 + |x|)^{3+\delta}} \quad - \infty < x < \infty.$$

We easily see that $\delta = \text{Sup} \{\varepsilon\colon \int x^{2+\varepsilon} f(x)\, dx < \infty\}$. By straightforward minimization techniques one can show for this class of densities that

$$\lim_{N\to\infty} \frac{\text{mse}}{\Delta^2/12} = 1 + \frac{2}{\delta}.$$

Property 8 is of interest because it sets forth a basic difference between uniform and nonuniform quantizers. For the nonuniform quantizer we can expect the mean square error to be of the order $1/N^2$. We can expect this rate of convergence to zero for the uniform quantizer only if the probability density is of finite support. We may show for the optimal uniform Gaussian quantizer that the error is the same or larger than $\ln N/N^2$.

REFERENCES

[1] B. Widrow, "Statistical analysis of amplitude quantized sampled data systems," Trans. Amer. Inst. Elec. Eng., Pt. 11, Applications and Industry, Vol. 79, pp. 555-568, Jan. 1960.

[2] H. Gish and J. N. Pierce, "Asymptotically efficient quantizing," IEEE Trans. Inform. Theory, Vol. IT-14, pp. 676-683, Sept. 1968.

[3] A. B. Sripad and D. L. Snyder, "A necessary and sufficient condition for quantization errors to be uniform and white," IEEE Trans. Acoustics, Speech, and Signal Processing, Vol. ASSP-25, pp. 442-448, Oct. 1977.

[4] P. Zador, "Development and evaluation of procedures for quantizing multivariate distributions," Ph.D. dissertation, Stanford University, Stanford, CA 1964.

[5] W. A. Pearlman and G. H. Senge, "Optimal quantization of the Rayleigh probability distribution," IEEE Trans. Communications, Vol. COM-27, pp. 101-112, Jan. 1979.

[6] J. A. Bucklew and N. C. Gallagher, Jr., "A note on optimum quantization," IEEE Trans. Info. Theory, Vol. IT-25, pp. 365-366, May 1979.

COMPLEXITY ASPECTS OF SYSTEM DIAGNOSIS

PIERO MAESTRINI
Istituto di Elaborazione dell'Informazione del CNR,
Pisa, Italy

ABSTRACT

Two relevant problems in the diagnostic model of Preparata, Metze and Chien consist in identifying classes of optimal sequentially diagnosable systems and in evaluating the complexity of syndrome decoding for sequential diagnosis. The problem of determining optimal sequentially diagnosable systems is still unsolved except for the cases when the diagnosability equals the smallest or the largest permissible value; and the known bounds to the complexity of optimal systems are supposed to be quite weak. Further, the problem of syndrome decoding is known to be NP-complete in the general case. A class of diagnostic systems is introduced, whose members attain all permissible values of sequential diagnosability. It is shown that the upper bound to complexity of optimal system established by such class is tighter than those previously known, and that complexity of syndrome decoding is $0(|V|)$, where $|V|$ is the number of units.

INTRODUCTION

This paper reconsiders some problems underlying the theory of system diagnosis. **It** is assumed that the system under consideration be partitioned into a number of units. Each unit is supposed to possess computational resources sufficient to enable it to test one or more of the remaining units. Each test is supposed to be complete for the class of relevant faults in the tested unit and each test outcome is supposed to be binary. The test outcome is fully significant only if the testing unit is fault-free. Different assumptions have been made in the literature to take into account the degree of test invalidation ensuing from the presence of faults in the testing unit; such assumptions result in different diagnostic models [2,3,4,5]. In this paper the original model due to Preparata, Metze and Chien [1] will be retained, and the test outcome will be considered completely unreliable if the testing unit is faulty.

This model consists of a directed graph $G=(V,A)$, where each vertex $v_i \in V$ corresponds to a unit of the system and there exists one arc (v_i,v_j) from vertex v_i to vertex v_j if and only if the unit represented by v_i tests the unit represented by v_j. Following to application of the test set each arc is labeled with the corresponding test outcome $s(v_i,v_j)$, where $s(v_i,v_j)=0$ if the test passes and $s(v_i,v_j)=1$ if the test fails. The set s of arc labelings resulting from an application of the test set is called a syndrome.

If all units which are faulty at the time of test execution are identified by a process of syndrome decoding, one-step diagnosis is said to occur. If the syndrome decoding enables identification of at least one faulty unit, the diagnosis is called sequential or with repair. For any given system both one-step and sequential diagnosis are possible for all syndromes provided that the number of faulty units does not exceed critical values, called the one-step diagnosability t_0 and sequential diagnosability t_r; such values are integers with $t_r \geq t_0 \geq 0$. The reader is referred to [1,2,3,4,6,7] for more details and results on sequential and one-step diagnosis of digital systems.

One problem of interest in system diagnosis consists in evaluating the complexity of self-diagnosable systems. A reasonable measure of the complexity is given by the number of arcs in the diagnostic model. Such measure needs to be related to the number $|V|$ of units and to the value of diagnosability, either one-step or sequential. Classes of optimal diagnosable systems (that is, diagnosable systems of minimal complexity) have been determined in the model of Barsi, Grandoni and Maestrini for all values of $|V|$ and for any permissible value of diagnosability, both one step and sequential [4], the same problem has been solved in the model of Preparata Metze and Chien limited to the case of one-step diagnosis [7]. The problem of determining optimal sequentially diagnosable systems in the model of Preparata, Metze and Chien is still unsolved, although optimal diagnosable systems have been found for any $n=|V|$ when $t_r=t_{r\ min}=\lceil 2\sqrt{n}\rceil-3$ [6] or $t_r=t_{r\ max}=\left\lfloor\frac{n-1}{2}\right\rfloor$ (1). An upper bound to the complexity of optimal sequentially diagnosable systems is also known for any permissible t_r [3]. A tighter bound will be established in this paper, by analysing a class of sequentially diagnosable systems whose diagnostic graph is a rosace [4].

Additional interest resides in evaluating the complexity of syndrome decoding, that is the computational effort required to identify the faulty units, or at least one of them, when the syndrome is given. It has been proved [9] that syndrome decoding belongs to the class of NP-complete problems, both in one-step and sequential diagnosis, although there may exist classes of diagnosable systems (e.g. the simple circuit in the case of sequential diagnosis) for which the syndrome decoding is computationally efficient. It will be shown that the class of sequentially diagnosable systems introduced in this paper is easily diagnosable, since the complexity of syndrome decoding is $O(|V|)$. The relevance of this result resides in the fact that the class under consideration contains sequentially diagnosable systems with arbitrary values of t_r and the complexity of the system itself reaches the best known upper bound to optimality.

SEQUENTIAL DIAGNOSABILITY

Let $G=(V,A)$ be the diagnostic graph of a given system and s be a syndrome. Subset $F\subseteq V$ is called a consistent fault pattern of s if $s(v_i,v_j)=1$ for all $(v_i,v_j)\in A$ such that $v_i\in V-F$, and $v_j\in F$ and $s(v_i,v_j)=0$ for all $(v_i,v_j)\in A$ such that $v_i\in V-F$, $v_j\in V-F$. Consider the set $\mathscr{F}_s$ of consistent fault patternsof s, such that $|F|\leq t$ for each $F\in\mathscr{F}_s$. If $\bigcap_{F\in\mathscr{F}_s} F\neq\emptyset$, then any unit in

---

(1) $\lceil x\rceil$ denotes the smallest integer not less that x; similarly $\lfloor x\rfloor$ is the greatest integer not greater than x.

$\bigcap_{F \in \mathscr{F}_s} F$ will be diagnosed as faulty under the hypothesis that the number of units does not exceed t. The greatest integer t_r such that this property holds for any syndrome s is the sequential diagnosability of the given graph [3].

Assuming, without any loss of generality, [4] that the diagnostic graph is strongly connected, the following ideas provide a constructive approach to determining the sequential diagnosability, or at least a lower bound to this parameter. Given a syndrome s and any $v \in V$, let $g_0^s(v)$ and $g_1^s(v)$ denote the minimal cardinalities of consistent fault patterns F_0 and F_1 of s such that $v \notin F_0$ and $v \in F_1$, respectively. Under the hypothesis that the number of faults does not exceed $x_s(v)=\max(g_0^s(v), g_1^s(v))-1$, v is unambiguously recognized to be faulty if $g_1^s(v) \leq x_s(v)$ and otherwise fault-free. In turn, identifying v as fault-free enables identification of at least one faulty unit under the hypothesis that the diagnostic graph is strongly connected, since any path beginning at v and consisting of h arcs labeled with 0 $(h \geq 0)$ followed by an arc (v_i, v_j) labeled with 1, will identify v_j as faulty.

Observe that, whenever $g_0^s(v)=g_1^s(v)$, the cardinality of every consistent fault pattern of syndrome s is greater than $x_s(v)$ and s cannot occur in the hypothesis of at most $x_s(v)$ faults. Also, every $v' \in V$ other than v, has $g_0^s(v')=g_0^s(v)$ and $g_1^s(v')=g_1^s(v)$ if $g_0^s(v')=g_1^s(v')$ and $\max(g_0^s(v'), g_1^s(v')) > g_0^s(v)$ otherwise.

Consider $t_r(s)= \max_{v \in V}(x_s(v))$ and let S be the set of all syndromes: the integer $t_r= \min_{s \in S}(t_r(s))$ is the sequential diagnosability. Further, assume that $x_s(v)=\max(g_0^s(v), g_1^s(v))-1$; $v \in V$, is known for all $s \in S$: then $t'_r= \min_{s \in S}(x_s(v))$ is a lower bound to sequential diagnosability t_r. Tighter bounds may be determined by considering the index $x_s(v)$ above defined for each $v \in V'$, where $V' \subseteq V$ and letting $t'_r(s)= \max_{v \in V'}(x_s(v))$ and $t'_r = \min_{s \in S}(t'_r(s))$.

For example, consider a diagnostic graph G = (V,A) consisting of a simple circuit of n vertices (Fig. 1) and, for any syndrome s, the unique [1] partition of the set V into sequences $(v_{i1}, v_{i2}, \ldots, v_{ik})$, $k \geq 2$, such that $s(v_{i(k-1)}, v_{ik})=1$

and $s(v_{i(q-1)},v_{iq})=0$ for $1<q<k$. Assuming that syndrome s partitions V into ν sequences and λ be the number of

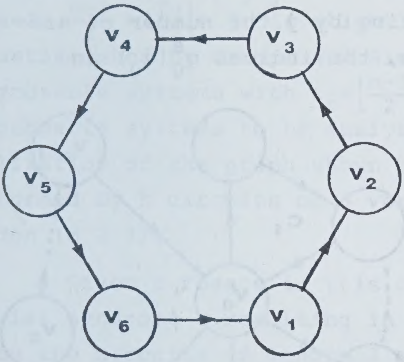


Fig. 1

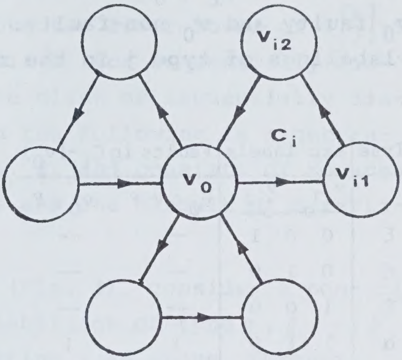


Fig. 2

vertices in a longest sequence, let v_f be the last vertex in a sequence of length λ. It is easily seen that $g_0^s(v_f)=\nu+\lambda-2$ and $g_1^s(v_f)=\nu$. The integer $t_r'=\lceil 2\sqrt{n}\rceil-3$, obtained by minimizing $x_s(v_f)=\max(g_o^s(v_f),g_1^s(v_f))-1$ over the set of permissible pairs (ν,λ) (that is, over the set S of syndromes), is a lower bound to sequential diagnosability of the simple circuit of n vertices. However, since $\lceil 2\sqrt{n}\rceil-3$ is the upper bound to sequential diagnosability of the simple circuit [6], t_r' actually coincides with the sequential diagnosability t_r.

A CLASS OF SEQUENTIALLY DIAGNOSABLE SYSTEMS

Consider a rosace consisting of k circuits, where each circuit has 3 vertices (Fig. 2). In order to determine the sequential diagnosability of this graph, the indices $g_0(v_0)$ and $g_1(v_0)$ will ve evaluated. For each circuit C_i, $1\leq i\leq k$, in the rosace consider the set of all possible arc labelings, listed in Table 1. If an arc labeling denoted type ξ occurs, the vertex whose incoming arc is labeled with 1 is necessarily faulty and sequential diagnosis is trivial. Thus, consideration will be limited to the non-trivial case of syndromes s resulting in arc labelings of type α, β, γ or δ in all circuits. For each such arc labeling, the minimum number of faults in the node subset $C_i-\{v_0\}$ is listed in Table 1 for both hypotheses of v_0 faulty and v_0 non-faulty. Denoting by j the number of arc-labelings of type j in the rosace, the indices $g_0^s(v_0)$ and

Type	Arc labels			Faults in $C_i-\{v_0\}$	
	v_{i1} $\rightarrow\!\circ\!\rightarrow\!\circ\!\rightarrow$ v_{i2}			$v_0 \in V-F$	$v_0 \in F$
ξ	0	0	1	--	--
ξ	0	1	0	--	--
ξ	1	0	0	--	--
α	1	1	0	1	1
α	0	1	1	1	1
β	1	0	1	2	0
γ	0	0	0	0	2
δ	1	1	1	2	1

Table 1

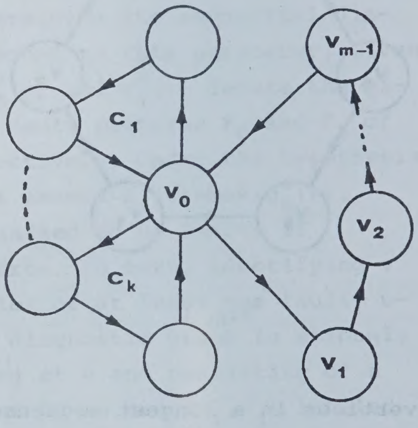


Fig. 3

$g_0^s(v_0)$ are immediately computed as follows:

$$g_0^s(v_0) = \alpha + 2\beta + 2\delta$$
$$g_1^s(v_0) = \alpha + 2\gamma + \delta + 1 \qquad (1)$$

By introducing $k=\alpha+\beta+\gamma+\delta$, expressions (1) become:

$$g_0^s(v_0) = k + \beta - \gamma + \delta = k + x + \delta$$
$$g_1^s(v_0) = k+1 - (\beta - \gamma) = k - x + 1 \qquad (2)$$

where $x=\beta-\gamma$ is an integer depending on the syndrome s.

Since $g_0^s(v_0) + g_1^s(v_0) = 2k + 1 + \delta$, assuming $g_i^s(v_0) \leq k$, $i \in \{0,1\}$, implies $g_{1-i}^s(v_0) \geq k+1$ for all syndromes. It follows $t_r' = \max(g_0(v_0), g_1(v_0)) - 1 \geq k$ and the rosace is at least k-sequentially diagnosable. On the other hand, the sequential diagnosability of the rosace shown in Fig. 2 cannot exceed k since $k = \left\lfloor \frac{n-1}{2} \right\rfloor$ coincides with the upper bound to sequential diagnosability [1], where $n=2k+1=|V|$.

The preceding result shows that the connection assignment of n nodes corresponding to the rosace of Fig. 2 yelds sequentially diagnosable systems whose diagnosability t_r reaches the upper bound of $\left\lfloor \frac{n-1}{2} \right\rfloor$, while the number of arcs, equal to $a=3.\ \frac{n-1}{2} = 3\left\lceil \frac{n}{2} \right\rceil - 3$, is slightly above the lower bound of $3\left\lceil \frac{n}{2} \right\rceil - 5$ established by Ciompi and Simoncini [8] for sequentially diagnosable systems with $t_r = \left\lfloor \frac{n-1}{2} \right\rfloor$. The class of sequentially diagnosable systems to be analyzed in the following is a generalization of the graph shown in Fig. 2, and consists of rosaces formed by k circuits of 3 vertices and one circuit of m vertices ($m \geq 3$).

Given a rosace in this class (Fig. 3), consider a non-trivial syndrome s resulting in arc labelings of type α, β, γ, δ in the circuits of length 3 and define x as above. Further, assume that the circuit of length m has ν sequences and λ is the maximum length of such sequences. Using the same techniques described in the preceding examples, the indices $g_0^s(v_0)$ and $g_1^s(v_0)$ are evaluated as follows:

1) v_0 is the last vertex in a sequence of length λ :
   $g_0^s(v_0) \geq k+x+\nu+\lambda-2$; $g_1^s(v_0) = k-x+\nu$
2) v_0 is the last vertex in a sequence of length $\ell \leq \lambda$:
   $g_0^s(v_0) \geq k+x+\nu+\lambda-2-(\lambda-\ell)$; $g_1^s(v_0) = k-x+\nu$
3) v_0 is not the last vertex in a sequence: $g_0^s(v_0) \geq k+x+\nu$;
   $g_1^s(v_0) = k-x+\nu$,

where the inequality in expressions for $g_0^s(v)$ results from neglecting δ. Let $t_r'(s)$ be an estimate of the maximum number of faults under which the state of at least one unit can be diagnosed when syndrome s occurs. If $g_0^s(v_0) > t_r'(s)$ $\{ g_1^s(v_0) > t_r'(s) \}$, unit v_0 is diagnosed as faulty {fault-free} in the hypothesis of $t_r'(s)$ or less faults, the case where $g_0^s(v_0) > t_r'(s)$ and $g_1^s(v_0) > t_r'(s)$ corresponding to the occurrence of more than $t_r'(s)$ faults. If $g_0^s(v_0) \leq t_r'(s)$, $g_1^s(v_0) \leq t_r'(s)$ the state of unit v_0 cannot be diagnosed; however denoting by v_f the last unit in a sequence of length λ it is easily seen that in this hypothesis $g_0^s(v_f) = 2k+2\nu-t_r'(s)+\lambda-2$ in case 1 and 3 and $g_0^s(v_f) = 2k+2\nu-t_r'(s)+\lambda-2+\ell-2 \geq 2k+2\nu-t_r'(s)+\lambda-2$ in case 2, where the last inequality derives from $\ell \geq 2$ [1] . In order v_f to be diagnosed as faulty in the assumption of $t'_r(s)$ or less faults, must be $t'_r(s) \le g_0(v_f)-1$ and $t'_r(s)$ is determined from the preceding expressions as $t'_r(s)=k+\nu+\left\lfloor\frac{\lambda-3}{2}\right\rfloor$. A lower bound t'_r to sequential diagnosability is thus determined by minimizing $t'_r(s)$ over the set of permissible pairs (ν, λ), resulting from consideration of all syndromes, as stated by the following theorem whose proof is omitted for the sake of brevity.

Theorem 1. A lower bound to sequential diagnosability of the rosace consisting of k circuits of length 3 and one circuit of length m is

$$t'_r=k+\left\lceil\frac{\sqrt{2m+1}-1}{2}\right\rceil+\left\lfloor\frac{1}{2}\left(\left\lceil\frac{m}{\left\lceil\frac{\sqrt{2m+1}-1}{2}\right\rceil}\right\rceil-3\right)\right\rfloor$$

The class of design corresponding to rosaces of Fig. 3 includes sequentially diagnosable systems whose diagnosability spans the values between the lower and the upper bound. It is interesting to observe that while the rosaces of Fig. 3 correspond to optimal sequentially diagnosable systems in the model of Barsi, Grandoni and Maestrini [4], this property does not hold in the model of Preparata, Metze and Chien. This can be verified by observing that the rosace with m=5, where the number of vertices is n=2k+5 and the number of arcs is a=3k+5, has $t_r = k+2=\left\lfloor\frac{n-1}{2}\right\rfloor$ and thus reaches the upper bound to sequential diagnosability, while the number of arcs exceeds by one the lower bound of $3\left\lceil\frac{n}{2}\right\rceil-5$ holding for optimal designs with the same diagnosability.

The problem of determining an upper bound to the complexity of optimal sequentially diagnosable systems with arbitrary $|V|$ and any permissible value of t_r has been considered by Maheshwari and Hakimi [3]. They have established a bound of $n+t_r-1$ for t_r-diagnosable systems whose diagnostic graph has n vertices. Since in the rosace of Fig. 3 the number of arcs is a=3 k+m and n=2 k+m, $k=t'_r-\left\lceil\frac{\sqrt{2m+1}-1}{2}\right\rceil-\left\lfloor\frac{1}{2}\left(\left\lceil\frac{m}{\left\lceil\frac{\sqrt{2m+1}-1}{2}\right\rceil}\right\rceil-3\right)\right\rfloor$, it is easily seen that the complexity of a system whose diagnosability is no less than t'_r is equal to $a=n+t'_r-\left\lceil\frac{\sqrt{2m+1}-1}{2}\right\rceil-\left\lfloor\frac{1}{2}\left(\left\lceil\frac{m}{\left\lceil\frac{\sqrt{2m+1}-1}{2}\right\rceil}\right\rceil-3\right)\right\rfloor$, which constitutes a tighter bound to optimality.

SYNDROME DECODING

Another problem of interest consists in evaluating the computational effort required to actually diagnose at least one unit, given the syndrome. It has been proved that syndrome decoding in sequential diagnosis is a NP-complete problem [9] and this implies that sequential diagnosis becomes computationally untractable as the size of the problem increases. However efficient decoding algorithms may exist for special classes of diagnostic graphs: this has been actually proved for the simple circuit of Fig. 1 [9] . A similar result holds for the k-rosace shown in Fig. 3, since it is clear that the following decoding algorithm, which derives from the preceding analysis of the diagnostic properties of the k-rosace, is $0(|V|)$ in time. With the notations defined above, the decoding algorithm is as follows:

A1) If any one of the circuits of length 3 has the arc labeling of type ξ and arc (v_{ij}, v_{ik}) is labeled with 1, vertex v_{ik} is recognized to be faulty. Else:

A2) Determine if v_0 is the last element of a sequence in the circuit of length m and if such sequence has legth λ , and compute $g_0^s(v_0)$, $g_1^s(v_0)$ using the corresponding formulae. If $g_0^s(v_0) > t'_r$, v_0 is recognized to be faulty in the hypothesis of at most t'_r faults; if $g_1^s(v_0) > t'_r$, v_0 is diagnosed as fault-free in the same hypothesis. Else:

A3) Denoting by v_f the last vertex in a sequence of length m, v_f is recognized to be faulty in the hypothesis of at most t'_r faults.

REFERENCES

[1] F.P. Preparata, G. Metze and R.T. Chien, "On the Connection Assignment Problem of Diagnosable Systems". IEEE Trans. Electron. Comput., Vol. EC-16, pp. 848-854, Dec. 1967.

[2] J.D. Russel and C.R. Kime, "System Fault diagnosis:

Closure and Diagnosability with Repair". IEEE Trans. on Comput., Vol. C-24, pp. 1078-1089, Nov. 1975.

[3] S.N. Maheshwari and S.L. Hakimi, "On Models for Diagnosable Systems and Probabilistic Fault Diagnosis". IEEE Trans. on Comp., Vol. C-25, pp. 228-236, March 1976.

[4] F. Barsi, F. Grandoni and P. Maestrini, "A Theory of Diagnosability of Digital Systems". IEEE Trans. on Comput., Vol. C-25, pp. 585-593, June 1976.

[5] L. Simoncini and A. Friedman, "Incomplete Fault Coverage in Modular Multiprocessor Systems". Proc. ACM Annual Conference 1978 (Washington D.C., 1978, pp. 210-216.

[6] F.P. Preparata, "Some Results on Sequentially Diagnosable Systems". Proc. Hawaii Int. Conference on System Sciences; Univ. of Hawaii Press, Jan. 1968, pp. 623-626.

[7] S.L. Hakimi and A.T. Amin, "Characterization of Connection Assignments of Diagnosable Systems". IEEE Trans. on Comput., Vol. C-23, pp. 86-87, Jan. 1974.

[8] P. Ciompi and L. Simoncini, "Analysis and Optimal Design of Self-Diagnosable Systems with Repair". IEEE Trans. on Comput. Vol. C28 pp. 362-365, May 1979.

[9] H. Fujiwara and K. Kinoshita, "On Computational Complexity of System Diagnosis". IEEE Trans. on Comput., Vol. C-27, pp. 379-384, April 1978.

ON FAULT DIAGNOSIS OF ITERATIVE LOGIC ARRAYS*

R. Parthasarathy and S. M. Reddy
Electrical and Computer Engineering
University of Iowa, Iowa City, Iowa 52242

ABSTRACT

Fault Detection and Location in one-dimensional Iterative Logic Arrays of combinational cells with and without directly observable outputs from each cell are studied. Methods of modifying the basic cell flow-table so as to facilitate fault detection are given. Improvements over previously reported methods and an error in an earlier work are pointed out.

I. INTRODUCTION

Due to advances in semiconductor technology, it has become possible to fabricate large integrated semiconductor arrays containing hundreds or thousands of gates. An attractive realization of these arrays is in a cellular form because of its structural simplicity, ease of fabrication, ease of circuit and logic design and simplified testing and diagnosis.

The testing procedure consists of applying test inputs to the accessible input lines of the array and observing the outputs at the accessible output lines. These lines are usually associated with the boundaries of the array, but some input and output lines of each cell in the array may be additionally accessible.

The problem of fault detection and location in Iterative Logic Arrays (ILA) of combinational cells was first studied by Kautz [2]. It was assumed that all possible cell inputs must be applied to each cell in order to test it completely and that the fault in a cell may affect the cell outputs in any arbitrary manner.

In general, the number of tests required to test a one-dimensional array of p cells is a function of p and is at most pmn(n-1) where m is the number of comumns and n is the number of rows of the cell flow-table. Kautz [2] has shown that an array can be tested with mn tests if and only if every column of the flow-table is a permutation column, i.e., each state appears exactly once as a next state entry in each column.

Friedman and Menon [1, 3, 4] have studied the same problem under restricted fault assumptions. Under these assumptions, a cell can be tested completely by applying a specified subset of the set of all cell inputs and each fault will affect the cell outputs in a particular known manner. Knowledge of actual realization of the cell is implicit in these assumptions. Necessary and sufficient conditions for testability of one-dimensional arrays were derived [3,4].

The problem of testing a finite but arbitrarily long, one-dimensional array of p cells with a fixed constant number of tests independent of p has also been studied [5-7]. Such arrays are referred to as C-testable [5].

---

*The research reported has been supported by the Air Force Office of Scientific Research Grant No. AFOSR-78-3582.

In this paper, we consider certain necessary and sufficient conditions on the cell flow-table to facilitate fault detection in one-dimensional ILAS of combinational logic, achieving a small test set in the process. Atmost one cell in the array is assumed to be faulty, and the fault may cause the output of the faulty cell to change in any arbitrary manner, but the cell still realizes combinational logic. The testing process involves verification of each of the mn transitions in the flow-table and hence does not require any knowledge of actual realization of the cell. The main idea in this paper is the introduction of a testing strategy that we call *one-step testability*.

II. PRELIMINARIES

Fig. 1 shows a one-dimensional ILA with the notation to be used. The signal flow is from left to right. A typical cell receives the input x from its left-hand neighbour and an external input z. It generates an external output $\hat{z}$ and transmits an output $\hat{x}$ to its right-hand neighbour. The accessible inputs consist of the x input to the leftmost cell and the z inputs to all cells in the array. The accessible outputs are the $\hat{x}$ output of the rightmost cell and all $\hat{z}$ outputs. Subscripts are used to refer to specific inputs or outputs.

The behavior of a cell can be described by a flow-table which has a row for each x input (called a present state) and a column for each z input. The entries in the table consist of pairs of cell outputs (called next state and output) $(\hat{x}, \hat{z})$, for each combination (x, z) of cell inputs. Thus, the entry in row x_i, column z_j is represented as $\hat{x}(x_i, z_j)$, $\hat{z}(x_i, z_j)$. *We assume that each present state appears at least once as a next state entry in the cell flow-table.*

Definition 1: An array is said to be *testable* for a fault f_i if there exists a test t_i that facilitates the effect of this fault to be observable at an accessible output line of the array.

Definition 2: Let $X = \left\{x_1, x_2, \ldots, x_i, \ldots, x_n\right\}$ be the set of present states of the cell flow-table. For every $x_i \in X$, let $S_i \triangleq \left\{x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n\right\} = X - \left\{x_i\right\}$.

Note that if all the states in X are applicable to the leftmost cell of the array, then all of them are applicable to any arbitrary cell in the array, because each state in X appears at least once as a next state entry in the flow-table.

Consider the transition from a present state x_i to the next state x_j under the input z_k. We represent this by $x_i \overset{z_k}{} x_j$. If the next state reached under z_k from the present state x_i is x_r instead of x_j, $x_r \neq x_j$, we represent this fault by $x_i \overset{z_k}{} x_j/x_r$.

Consider the transition $x_i \overset{z_k}{} x_j$. If the next state reached is any of the (n - 1) faulty states in S_j, we represent this set of faults by $x_i \overset{z_k}{} x_j/S_j$. Thus, $x_i \overset{z_k}{} x_j/S_j$ represents all possible faults in the transition $x_i \overset{z_k}{} x_j$.

Definition 3: A transition $x_i^{z_k}\, x_j$ in cell C_r of the array is said to be one-step testable if there exists at least one input vector Z that can be applied to the cells $C_{r+1}, C_{r+2}, \ldots, C_p$ of the array such that the observable outputs of the array are different when x_j and any member of S_j are applied to the cell C_{r+1} of the array. Thus, a single test is both necessary and sufficient to test for the fault $x_i^{z_k}\, x_j/S_j$, if the transition $x_i^{z_k}\, x_j$ is one-step testable.

Definition 4: A cell is one-step testable if all mn transitions of the cell flow-table are one-step testable.

Definition 5: A one-dimensional ILA of arbitrary length p is said to be one-step C-testable if every cell of the array is one-step testable and if the entire array can be tested with a fixed constant number of tests independent of p.

Note that if an array is one-step C-testable, then it is also C-testable.

III. FAULT DETECTION

3.1 ILA Without $\hat{z}$ Outputs

Let $V = \left\{ z_1, z_2, \ldots, z_i, \ldots, z_m \right\}$ be the set of z inputs to any cell and assume that only the $\hat{x}$ outputs of the right most cell are observed.

Theorem 1: If all x inputs are applicable to the leftmost cell of the array considered, any arbitrary cell in the array is one-step testable if and only if for every $x_i \in X$, there exists atleast one $z_i \in V$ such that

$$\hat{x}(x_i, z_i) \neq \hat{x}(x_j, z_i) \text{ for every } x_j \in S_i.$$

Proof:

Necessity: Assume that the cell is one-step testable and for some $x_k \in X$, there does not exist any $z \in V$ satisfying the condition of the Theorem. Then there exists some $x_{\ell_t} \in S_k$ such that $\hat{x}(x_k, z_t) = \hat{x}(x_{\ell_t}, z_t)$ for every $z_t \in V$. Since all the states in X appear as next state entries in the flow-table, there exists at least one x_m and at least one z_r such that

$$\hat{x}(x_m, z_r) = x_k.$$

Without loss of generality, let us assume that the leftmost cell is tested for the transition $x_m^{z_r}\, x_k$. By our assumption $x_m^{z_r}\, x_k$ is one-step testable and there exists a vector Z that can be applied to the cells $c_2, c_3, \ldots c_p$ to test for the fault $x_m^{z_r}\, x_k/S_k$. Let z_t be the input in Z that is applied to c_2. But we know that there exists an $x_{\ell_t} \in S_k$ such that $\hat{x}(x_k, z_t) = \hat{x}(x_{\ell_t}, z_t)$ which implies that Z does not propagate the fault $x_m^{z_r}\, x_k/x_{\ell_t}$. Hence, the transition $x_m^{z_r}\, x_k$ is not one-step testable.

Sufficiency: Since the condition in the Theorem is satisfied, there exists a z_i for every $x_i \in X$ such that $\hat{x}(x_i, z_i) = x_j$ and $\hat{x}(S_i, z_i) = A_r \subseteq S_j$. x_j and any element in A_r can in turn be propagated by another input z_j since the condition is satistied for every $x_i \in X$. Thus, there exists a single input vector $Z = z_i z_j z_k, \ldots$ which propagates the fault $x_r{}^{z_t}\, x_i/S_i$ to the rightmost observable output of the array; i.e., any transition $x_r{}^{z_t}\, x_i$ in an arbitrary cell and, hence, the cells are one-step testable.

Corollary 1: Any arbitrary cell flow-table can be made one-step testable by adding a column so that it contains all the n states of X.

The following necessary and sufficient condition for one-step C-testability of ILA's can be proved

Theorem 2: An ILA of arbitrary length p is one-step C-testable if and only if for every $x_i \in X$ and $z_k \in V$, such that $\hat{x}(x_i, z_k) = x_j$, implies there exists a sequence R of z inputs such that (i) $\hat{x}\,(x_j, z_k R) = x_i$ and (ii) $\hat{x}\left(x_i, (Rz_k)^*\right) \neq \hat{x}\left(x_t, (Rz_k)^*\right)$, for every $x_t \in S_j$, where $(R\, z_k)^*$ is a sequence $Rz_k Rz_k Rz_k, \ldots$ of arbitrary length.

Definition 6: An input z_j of a cell flow-table having n states is said to be a strongly connecting input if z_j connects all the n states of the flow-table together in a single loop.

A sufficient condition for the existence of a strongly connecting input is that for some z_j, $\hat{x}(x_i, z_j) = x_{i+1} \pmod{n}$; $1 \leq i \leq n$, or $\hat{x}(x_i, z_j) = x_{i-1} \pmod{n}$; $1 \leq i \leq n$.

Definition 7 [3]: A pair graph of the cell flow-table is a directed graph whose nodes are all ordered pairs of states of the flow-table. There is a directed edge labeled with z_k from a node (x_i, x_j) to a node (x_k, x_ℓ) in the graph if and only if there is some $z_k \in V$ such that $\hat{x}(x_i, z_k) = x_k$ and $\hat{x}(x_j, z_k) = x_\ell$.

Note that a pair graph of a flow-table with n states consists of $n(n - 1)$ nodes.

Definition 8: The augmented pair graph of the cell flow-table consists of $n(n - 1)$ nodes representing all ordered pairs of states and n nodes representing $(x_1, S_1), (x_2, S_2), \ldots, (x_i, S_i), \ldots, (x_n, S_n)$. A directed edge with label z_k is drawn from node (a, b), a and b $\in X$, to node (x_i, S_i) iff $\hat{x}(a, z_k) = x_i$ and $\hat{x}(b, z_k) = x_j \in S_i$. A directed edge with label z_k is drawn from (x_i, S_i) to (x_j, S_j) iff $\hat{x}(x_i, z_k) = x_j$ and for every $x_t \in S_i$, $\hat{x}(x_t, z_k) = x_r \in S_j$ such that if $x_t, x_\ell \in S_i$ and $x_t \neq x_\ell$ implies $\hat{x}(x_t, z_k) \neq \hat{x}(x_\ell, z_k)$. A directed edge with label z_k is drawn from (x_i, S_i) to (c, d) iff $\hat{x}(x_i, z_k) = c$ and for every $x_j \in S_i$, $\hat{x}(x_j, z_k) = d$.

In the following discussion, $\hat{x}(x_k/x_r, z_t) = a/b$ is an abbreviation for $\hat{x}(x_k, z_t) = a$ and $\hat{x}(x_r, z_t) = b$. Next we give a set of sufficient conditions for one-step C-testability of ILAs.

Theorem 3: An ILA is one-step C-testable if a) the pair graph of the cell flow-table is strongly connected, b) the flow-table has no constant columns, c) there exists at least one strongly connecting input z_r and d) there exists at least one input z_k such that for some $x_k \in X$ and for every $x_r \in S_k$, $\hat{x}(x_k/x_r, z_k) = a/b$; $a, b \in X$.

Proof: Consider an arbitrary transition $x_i^{z}x_j$ in an arbitrary cell of the array. Since the flow-table does not contain constant columns, there exists an $x_m \in X$ such that $\hat{x}(x_i/x_m, z) = x_j/x_\ell$; $x_\ell \in S_j$.

Hence, in the augmented pair graph, there exists an edge labeled with z from the node (x_i, x_m) to the node (x_j, S_j). Since there exists at least one strongly connecting input z_r, there is a unique sequence of inputs $(z_r z_r z_r, \ldots, z_r)$ of length $< n$ that connects the node (x_j, S_j) to the node (x_k, S_k). Under condition d) there exists an input z_k that connects (x_k, S_k) and (a, b).

Since the pair graph is strongly connected, there exists an input sequence R that connects (a, b) to (x_i, x_m). Thus, there exists an input sequence $z\ z_r z_r, \ldots, z_r z_r R$ that connects the node (x_i, x_m) to itself. The length of this sequence is atmost n^2, as the augmented pair graph has n^2 nodes.

Thus, the input combination (x_i, z) can be applied to every qth cell $(q \leq n^2)$ in the array by a single test and the transition $x_i^{z}x_j$ is one-step testable and every cell in the array can be tested for the transition $x_i^{z}x_j$ by q tests which is independent of the length of the array. Hence, the ILA is one-step C-testable.

Consider an arbitrary cell flow-table that does not satisfy any of the conditions of Theorem 3. Add three columns, z_{m+1}, z_{m+2} and z_{m+3} and a row x_{n+1} to this table with entries as shown in Fig. 2.

The additional row x_{n+1} with the entries shown ensures that there are no constant columns. Columns z_{m+1} and z_{m+2} ensure that the pair graph is strongly connected. z_{m+1} creates $(n - 1)$ disjoint loops, each having n nodes. The assignment in z_{m+2} connects all these $(n - 1)$ loops in a single loop.

Also, z_{m+3} satisfies condition d) of Theorem 2. Thus, the columns z_{m+1}, z_{m+2} and z_{m+3} together satisfy all conditions of Theorem 2.

Corollary 2: Any arbitrary cell flow-table can be modified to make the corresponding ILA one-step C-testable by adding, at most, three columns and a row.

It should be pointed out that the pair graph of any arbitrary flow-table cannot be made strongly connected by the addition of the two columns z_{m+1} and z_{m+2} shown in Fig. 3 as reported earlier [5]. For example, consider the flow-table shown in Fig. 4(a) with $n = 4$. The associated pair graph is shown in Fig. 4(b).

In this graph, it is not possible to go to the middle loop from any of the other two loops.

The above observation is true of any flow-table for which $n > 3$, with the assignments shown in columns z_{m+1} and z_{m+2} of Fig. 3.

The earlier method of modifying cell flow-tables to make them C-testable required $[\log_2 n(\log_2 n + 1)/2] + 2$ additional columns for any arbitrary cell flow-table [5]. The proposed method requires 3 additional columns for any arbitrary flow-table so that the corresponding array is one-step C-testable.

To derive bounds on the number of tests required, assume an arbitrary cell flow-table. Add a row and three columns to this table as shown in Fig. 2. This makes the flow-table one-step C-testable. Since columns z_{m+1} and z_{m+2} of the modified flow-table contain all the n+1 states, only $2(n + 1)$ tests are required to test for all the transitions in these two columns in all the cells of the array.

The augmented pair graph of the flow-table has $(n + 1)^2$ nodes. Hence, any loop corresponding to the one-step C-testable transition $x_i^{z}\,x_j$ can contain atmost $(n + 1)^2$ nodes. Thus, atmost $(n + 1)^2$ tests are required to test for the transition $x_i^{z}\,x_j$ in all the cells of the array. Hence, an upper bound on the number of tests $|T|$ required to completely test the array is $|T| \leq (n + 1)^3\ (m + 1) + 2(n + 1)$.

3.2 ILA's with $\hat{z}$ Outputs: It has been shown that any given ILA can be made C-testable by making the cell flow-table reduced with respect to the $\hat{z}$ outputs and have repeatable transitions only [7]. It was shown in [7] that an additional column z_{m+1} containing all the n-states in such a manner that all the n-states are strongly connected in a single loop is necessary and sufficient to make any flow-table a repeatable transition flow-table.

From the arguments given earlier, ILA's of such cells are one-step C-testable. Let Q be an input sequence obtained by repeating z_{m+1} r times. It was shown that, for any n, the output entries in the column z_{m+1} can be chosen so that Q is a distinguishing sequence of optimum length

$$r = \lceil \log_2 n \rceil.$$

An upper bound on the number of tests T required to detect faults in arrays of such cells was given to be $|T| \leq mn(r + n) + n$.

Test lengths can be further reduced if the number of z outputs is larger than 2. An example of z_{m+1} for n = 8 is given in Fig. 5. We will refer to cells with cell tables that include such a z_{m+1} column as Type A cells.

IV. FAULT LOCATION

Location of faulty cells in ILAs was earlier studied [1-4]. Methods to augment arbitrary cell flow tables to allow location of a faulty cell to a pair of consecutive cells were given [4]. In this paper, we derive certain sufficient conditions on cell tables that will allow location of faulty cells to a pair of consecutive cells. We consider the case of cells with observable $\hat{z}$ outputs = 2. The following theorem can be proved.

Theorem 4: A faulty cell in an ILA can be located to a pair of consecutive cells if:

(i) each cell is a Type A cell,

(ii) there exists an input z such that $\hat{x}(x_i, z) = a$, for every $x_i \in X$, and

(iii) for every $x_i, x_j \in X$, $x_i \neq x_j$, there exists an input z_k such that $\hat{z}(x_i, z_k) \neq \hat{z}(x_j, z_k)$.

An arbitrary cell table can be augmented by adding $\max(2, \lceil \log_2 n \rceil)$ columns such that the conditions of Theorem 4 are met. The basic idea behind the augmentation is to observe that condition (iii) of Theorem 4 can be met by choosing the $\hat{z}$ outputs for the added columns such that the entries in each one of the n rows in these columns are unique. The $\hat{x}$ outputs in the added columns can be chosen such that conditions (i) and (iii) of Theorem 4 are satisfied as long as at least two columns are added. An example of such augmentation is given in Fig. 6, for n = 8.

Corollary 3: An arbitrary cell table can be augmented by adding at most $\max(2, \lceil \log_2 n \rceil)$ such that the resulting cell satisfies the conditions of Theorem 4.

An upper bound on the number of tests required to locate a faulty cell in ILAs satisfying Theorem 4 can be derived. Let $r = \lceil \log_2 n \rceil$ and $m' = m + \max(2, r)$, then the number of tests, $|T_L|$, required to locate a faulty cell to a pair of consecutive cells is,

$$|T_L| \leq (n+r+1)\,[(m'-1)n+2]$$

Note that $|T_L|$ is independent of the length of the array. Hence, we refer to ILAs of cells satisfying Theorem 4 as C-fault-locatable ILAs.

REFERENCES

[1]. A. D. Friedman and P. R. Menon, "Fault Detection in Digital Circuits," Prentice-Hall, Inc., 1971.

[2]. W. H. Kautz, "Testing for Faults in Cellular Logic Arrays," Proceedings of the 8th Annual Symposium on Switching and Automata Theory, 161-174, 1967.

[3]. P. R. Menon and A. D. Friedman, "Fault Detection in Iterative Logic Arrays," IEEE Transactions on Computers, C-20, 524-535, 1971.

[4]. A. D. Friedman and P. R. Menon, "Fault Location in Iterative Logic Arrays," Theory of Machines and Computations, ed. Z. Kohavi and A. Paz, Academic Press, 1971.

[5]. A. D. Friedman, "Easily Testable Iterative Systems," IEEE Transactions on Computers, C-22, 1061-1064, 1973.

[6]. F. J. O. Dias, "Truth-table Verification of an Iterative Logic Array," IEEE Transactions on Computers, C-25, 605-613, 1976.

[7]. T. Sridhar and J. P. Hayes, "Testing Bit-Sliced Microprocessors," Digest of Papers, Ninth Annual International Symposium on Fault-Tolerant Computing, 211-218, 1979.

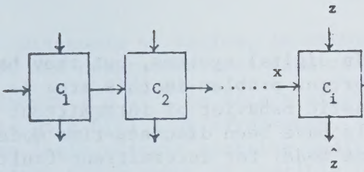

z
C1
C2
x
C_i
x̂
ẑ

Fig. 1

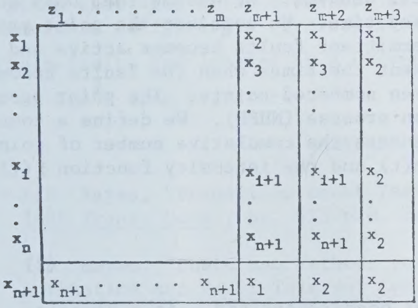

z1 . . . . zm
zm+1
zm+2
zm+3
x1
x2
xi
xn
xn+1
x2
x3
xi+1
xn+1
x1
x3
xn+1
x1
xn
x2
x1
x2
xn+1
x1
x2
x2

Fig. 2

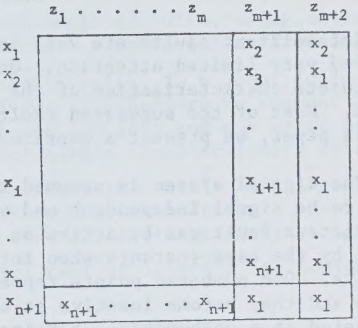

z1 . . . . zm
zm+1
zm+2
x1
x2
xi
xn+1
x2
x3
xi+1
xn+1
x1
x2
x1
x1
x1
x1

Fig. 3

	$z_1 \ldots z_m$	z_{m+1}	z_{m+2}
1		2	2
2		3	1
3		4	1
4		1	1

Fig. 4(a)

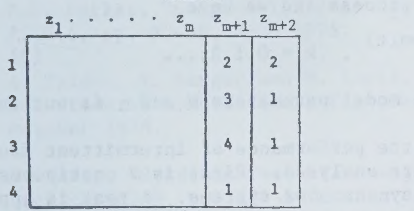

(1,2)
(2,3)
(1,3)
(2,4)
(1,4)
(2,1)
(4,1)
(3 4)
(4,2)
(3,1)
(4,3)
(3,2)

Fig. 4(b)

	$z_1 \ldots z_m$	z_{m+1}
A		B/0
B		C/0
C		D/1
D		E/1
E		F/1
F		G/0
G		H/1
H		A/0

Fig. 5

	$z_1 \ldots z_m$	z_{m+1}	z_{m+2}	z_{m+3}
A		B/0	A/0	D/0
B		C/0	A/1	D/1
C		D/1	A/1	D/1
D		E/1	A/1	D/0
E		F/1	A/0	D/1
F		G/0	A/1	D/0
G		A/1	A/0	D/0
H		A/0	A/0	D/0

Fig. 6

A POISSON PROCESS MODEL FOR INTERMITTENT FAULTS IN DIGITAL SYSTEMS

PRAMOD K. VARSHNEY
Electrical & Computer Engineering Department
Syracuse University
Syracuse, New York 13210

SUMMARY

Intermittent faults are very common in digital systems, but they have received very limited attention. One important problem in this area is an accurate characterization of the stochastic behavior of intermittent faults. Most of the suggested fault models have been discrete-time models. In this paper, we present a continuous-time model for intermittent faults.

The digital system is assumed to be irredundant. The faults are assumed to be signal independent and well behaved. We assume that only one intermittent fault can be active at any time. We consider the point process formed by the time instants when intermittent faults becomes active and inactive. Odd numbered points represent the times when the faults become active and they become inactive at even numbered points. The point process is modeled as a nonhomogeneous Poisson process (NHPP). We define a counting process $\{N(t), t \geq 0\}$ which represents the cumulative number of points by time t. The mean value function m(t) and the intensity function $\lambda(t)$ are

$$m(t) = M(1-e^{-\alpha t}) \tag{1}$$

$$\lambda(t) = M\, e^{-\alpha t} \tag{2}$$

where M is the average number of point occurrences and α is a constant. The process is an independent-increment process and we have

$$P\,\{N(t) = k\} = \frac{\{m(t)\}^k}{k\,!}\, e^{-m(t)}, \quad k = 0,1,2,\ldots \tag{3}$$

An approach to the estimation of the model parameters M and α is outlined.

The model is used to determine the performance of intermittent fault detection procedures. Two schemes are analyzed. First is a continuous strategy which can be used to test asynchronous systems. A test is applied continuously and if no fault appears during the test application T, the circuit is declared to be fault free. The performance is described in terms of the probability of miss, Q, which is the probability that an intermittent fault is not detected during the test. For continuous strategy.

$$Q = \exp\,\{-M(1-e^{-\alpha T})\} \tag{4}$$

The other scheme is repetitive testing which is used for synchronous circuits. A test is applied every t' seconds and a maximum number of test repetitions R is specified. If no fault appears in R repetitions, the circuit is declared to be fault free. For this scheme.

$$Q = \prod_{i=1}^{R} \frac{1}{2}\,(1 + \exp\,\{-2M\,(e^{-\alpha t_{i-1}} - e^{-\alpha t_i})\}) \tag{5}$$

Some numerical results are also presented.

FAULT DIAGNOSIS BY COMPRESSED TEST DATA[*]

SUDHAKAR M. REDDY
Electrical and Computer Engineering
The University of Iowa, Iowa City, Iowa 52242

SUMMARY

Diagnosis of faults, in combinational logic circuits, using universal tests and compressed test data is considered. A universal test sequence of length $\approx n\,2^{n-1}$ to detect single stuck-at-faults, in n input logic circuits with $O(n)$ long signatures, is derived. A procedure to augment a given logic circuit with the addition of one AND gate and one extra output is given which makes it possible to detect single stuck-at-faults by universal tests of length 2^n and $O(n)$ long signatures. $O(n^2)$ long signatures are given that will allow detection of up to 3 stuck-at-faults. Next a procedure to design logic circuits such that all distinguishable single stuck-at-faults are located by a universal test of length 2^n in conjunction with an $O(n^2)$ long signature.

REFERENCES

1. J.P. Hayes, "Transition Count Testing of Combinational Circuits," IEEE Trans. Comp., pp. 613-620, June 1976.

2. J.P. Hayes, "Check Sum Methods for Test Data Compression," J. Design Automation and Fault-Tolerant Computing, pp. 3-17, October 1976.

3. K.P. Parker, "Compact Testing: Testing with Compressed Data," Proc. FTCS-6, pp. 93-98, June 1976.

4. A. Tzidon, I. Berger and M. Yoeli, "A Practical Approach to Fault Detection in Combinational Circuits," IEEE Trans. Comp., pp. 968-971, October 1978.

5. J. Savir, "Syndrome-Testable Design of Combinational Circuits," Digest FTCS-9, pp. 137-140, June 1979.

6. S.C. Seth, "Data Compression Techniques in Logic Testing: An Extension of Transition Counts," J. DA and FTC, pp. 99-114, February 1977.

7. J.P. Hayes, "Generation of Optimal Transition Count Tests," IEEE Trans. Comp., pp. 36-41, January 1978.

---

[*] The research reported has been supported by Air Force Office of Scientic Research Grant No. AFOSR-78-3582.

FAULT DIAGNOSIS ALGORITHMS FOR ASYMETRIC MODULAR ARCHITECTURES

G. G. L. MEYER
Electrical Engineering Department
The Johns Hopkins University
Baltimore, Maryland 21218

ABSTRACT

This paper is devoted to the analysis of algorithms for the automatic fault diagnosis of asymetric modular networks. The network model used is the one proposed by Preparata, Metze and Chien, the faults are assumed to be solid, and the Hakimi-Amin t-diagnosability hypothesis is supposed to be satisfied.

SUMMARY

Consider a system S of n modules $U_0, U_1, \ldots, U_{n-1}$ and a testing interconnection assignment $E = \{(i,j) \mid U_i \text{ tests } U_j\}$. It is assumed that when (i,j) is in E, the test outcome $a_{i,j}$ of U_i testing U_j is $a_{i,j} = 0$ if U_i believes U_j to be nonfaulty and $a_{i,j} = 1$ if U_i believes U_j to be faulty. A complete set of test outcomes, i.e., an outcome $a_{i,j}$ for each (i,j) in E is called a syndrome. The diagnosis problem consists in identifying the set F^* of faulty modules from the knowledge of one of the possible corresponding syndromes. The problem is complicated by the fact that if (i,j) is in E, and U_i is nonfaulty, then $a_{i,j} = 0$ implies that U_j is nonfaulty, and $a_{i,j} = 1$ implies that U_j is faulty, but if U_i is faulty, then U_j may be nonfaulty or faulty regardless of the value of $a_{i,j}$.

Consistent Partition

Let (G,F) be a partition of S and suppose that (i) if U_i is in G and $a_{i,j} = 0$, U_j is in G; (ii) if U_i is in G and $a_{i,j} = 1$, U_j is in F; and (iii) if U_i is in F and $a_{j,i} = 0$, U_j is in F, then the partition (G,F) is consistent with the given syndrome. Given a syndrome, we say that a subset F of S is admissible with respect to the syndrome iff the partition $(S-F,F)$ is consistent with the syndrome. Clearly, the set of faulty modules F^* is admissible with respect to any of its corresponding syndrome. Unfortunately, given a syndrome there usually exist many sets F which are admissible (e.g., S is always admissible) and it is not possible to identify F^* without additional assumptions.

This research was supported by the Office of Naval Research under Contract N00014-75-C-1196.

Probabilistic Approach

We associate with each subset F of S a probability that F is faulty, i.e., there exists a map $p(\cdot)$ from all the subsets of S into the interval $[0,1]$ such that given F, the a-priori probability that F is faulty is simply $p(F)$. It is then reasonable to consider the following problem.

Problem 1: Given a syndrome, find an admissible set $F^{\#}$ such that $p(F^{\#}) \geq p(F)$ for all admissible sets F.

The solution of Problem 1 is therefore an admissible set which is also the most likely to occur. We still have a problem when the solution of Problem 1 is not unique, and α-probabilistic diagnosability hypotheses insure that the existence of an admissible set F such that $p(F) \geq \alpha$, imply that the solution of Problem 1 is unique and equal to F. It is difficult to evaluate $p(F)$ for a given set F, and it is often assumed that $p(F_1) < p(F_2)$ whenever $||F_1|| > ||F_2||$, where $||F||$ denotes the cardinality of F. In such a case, Problem 1 reduces to:

Problem 2: Given a syndrome, find an admissible set $F^{\#}$ such that $||F^{\#}|| \leq ||F||$ for all admissible sets F.

As with Problem 1, we may still have difficulty with the non-uniqueness of solution of Problem 2, and in this case, t-diagnosability hypotheses insure that the existence of an admissible set F such that $||F|| \leq t$ imply that the solution of Problem 2 is unique and equal to F. Thus, under such hypotheses, and with the assumption that $||F^*|| \leq t$, we know that the solution of Problem 2 will not only be unique, it will also be equal to F^*.

Fault Diagnosis Algorithms

Once the original problem of identifying F^* has been reduced to either Problem 1 or Problem 2, it remains the difficult task of producing algorithms to solve these problems. In this paper, we will concentrate our efforts mainly on Problem 2, when the t-diagnosability of Hakimi and Amin [5] is satisfied. In particular, we will discuss and compare algorithms which take into account the structure of the Hakimi-Amin hypothesis, with algorithms which do not make intensive use of this structure.

REFERENCES

[1] Allan, F. J., Kameda, T., and Toida, S., An Approach to the Diagnosability Analysis of a System, IEEE Transactions on Computers, Vol., October 1975, pp. 1040-1042.

[2] Barsi, F., Grandoni, F., and Maestrini, P., A Theory of Diagnosability of Digital Systems, IEEE Transactions on Computers, Vol. C-25, June 1976, pp. 585-593.

[3] Corluhan, A. M., and Hakimi, S. L., On an Algorithm for Identifying Faults in a T-Diagnosable System, Proceedings of the 1976 Conference On Information Science and Systems, The Johns Hopkins University, Baltimore, 1976, pp. 370-375.

[4] Fujiwara, H., and Kinoshita, K., Some Existence Theorems for Proba-Bililtically Diagnosable Systems, IEEE Transactions on Computers, Vol. C-27, No. 4, April 1978, pp. 379-384.

[5] Hakimi, S. L., and Amin, A. T., Characterization of Connection Assignment of Diagnosable Systems, IEEE Transactions on Computers, Vol. C-23, January 1974, pp. 86-88.

[6] Kameda, T., Toida, S., and Allan, F. J., A Diagnosing Algorithm for Networks, Information and Control, Vol. 29 (1975), pp. 141-148.

[7] Kime, C. R., An Analysis Model for Digital System Diagnosis, IEEE Transactions on Computers, Vol. C-19, No. 11, November 1970, pp. 1063-1073.

[8] Maheshwari, S. N., and Hakimi, S. L., On Models for Diagnosable Systems and Probabilistic Fault Diagnosis, IEEE Transactions on Computers, Vol. C-25, March 1976, pp. 228-236.

[9] Meyer, G. G. L., Fault Diagnosis of Modular Networks with a Small Number of Faults, Proceedings of the Fifteenth Annual Allerton Con-Ference on Communication, Control, and Computing, Allerton House, Monticello, Illinois, September 1977, pp. 727-731.

[10] Meyer, G. G. L., and Masson, G. M., An Efficient Fault Diagnosis Algorithm for Symmetric Multiple Processor Architectures, IEEE Transactions on Computers, Vol. C-27, No. 11, November 1978, pp. 1059-1063.

[11] Preparata, F. P., Metze, G., and Chien, R. T., On the Connection Assignment Problem of Diagnosable Systems, IEEE Transactions on Electronic Computers, Vol. EC-16, December 1967, pp. 848-854.

[12] Russell, J. D., and Kime, C. R., System Fault Diagnosis; Closure and Diagnosibility with Repair, IEEE Transactions on Computers, Vol. C-24, November 1975, pp. 1078-1089.

[13] Russell, J. D., and Kime, C. R., System Fault Diagnosis: Masking, Exposure, and Diagnosability without Repair, IEEE Transactions on Computers, Vol. C-24, No. 12, December 1975, pp. 1155-1161.

[14] Smith, J. E., Universal System Diagnosis Algorithms, IEEE Transactions on Computers, Vol. C-28, No. 5, May 1979, pp. 374-378.

COMPUTER SYSTEM DIAGNOSIS ASSUMING ASYMMETRIC INVALIDATION

C. S. HOLT and J. E. SMITH
University of Wisconsin, Madison, WI

ABSTRACT

This paper is concerned with system diagnosis through analysis of a set of diagnostic test results. It is assumed that a faulty unit may cause one or more tests on good units to fail but may not cause tests on faulty units to pass. A general system model is used in which each test may be performed by a set of units. Conditions for diagnosability with and without repair are determined. Exact methods, as well as simpler approximate methods, are proposed for determining diagnosability and for performing diagnosis.

I. INTRODUCTION

Due to the demand for higher reliability and availability in complex digital systems, the ability to rapidly and accurately locate system faults is becoming increasingly desirable. One way to accomplish fault location is to construct a set of diagnostic tests that can be analyzed to determine the faulty unit or units. Research in this area includes a number of theoretical studies exploring the relationships between faults, tests and diagnostic procedures. In these studies, a system is generally modeled as a set of nodes or units, each having an associated test or tests. Results usually fall in the following categories:

--Theoretical bounds on system diagnosability: Here, diagnosability of a system is a measure of the number of faulty units the system can tolerate without losing the capacity of being correctly diagnosed.

--Diagnosability analysis algorithms: Given a system description, an analysis algorithm computes the diagnosability of the system.

--Diagnosis algorithms: Given a system description and a set of test results, a diagnosis algorithm identifies the faulty unit or units.

--Optimal systems: In a diagnostic sense, optimal systems are ones that attain maximum diagnosability with a minimum number of tests or test applications.

Results have been derived for both "diagnosis without repair," in which all faulty units are identified by one application of the test set, and "diagnosis with repair," in which each application of the test set identifies at least one faulty unit.

Because different assumptions and system models have been used in the various studies, a variety of results have been obtained. In this work, diagnosis is studied under a combination of assumptions that is intended to be both realistic and workable. Some particularly important premises are those concerning "test invalidation," and system topology. Commonly used assumptions in these areas are reviewed below.

The analysis of diagnostic tests is complicated by the fact that some of the units performing tests may themselves be faulty and may therefore produce invalid test results. In some of the first serious work on diagnosability, Preparata et al. [1] introduced the assumption that tests performed by good units always give correst results, while tests performed by faulty units can produce any result. This straightforward assumption has been used in most of the subsequent work, and is referred to as symmetric invalidation. In contrast, Barsi et al. [2] used the assumption that faulty units always fail tests; that is, faults can only cause tests on good units to fail; they cannot cause tests on faulty units to pass. The motivation for this asymmetric invalidation assumption is discussed in a later section.

Preparata et al. [1] also used a model in which only one unit could invalidate a given test. We refer to this as "single mask per test" or SMPT. This assumption was relaxed in the "multiple mask per test" (MMPT) model proposed by Kime and used by Russell and Kime in slightly modified form [3,4]. Another special case that is often considered is systems with a single test per "fault" (i.e., per unit), abbreviated STPF. The general case of multiple tests per fault (MTPF) is more difficult.

In this paper the asymmetric invalidation of Barsi et al. [2] is combined with the general MTPF, MMPT system model proposed by Kime [5]. Bounds on system diagnosability are found, and methods for diagnosability analysis and diagnosis are proposed. Optimal systems are not covered since they would be the same as in the SMPT case [2].

A. Diagnostic Model

A system S is represented by a set $F = \{f_1,\ldots,f_n\}$ of potentially faulty units, and a set $T = \{t_1,\ldots,t_m\}$ of tests on these units. A "unit," as used here, is simply any portion of a system that is conveniently tested as a single entity. It is usually convenient for these "units" to correspond to sub-systems or physical units in the ordinary sense, but this need not always be the case. Any subset of F is called a fault pattern, and the set of fault patterns containing t or fewer faulty units (faults) is denoted F(t). Faults are assumed to be permanent.

Each test t_j is complete for exactly one unit f_i; that is, t_j always fails when f_i alone is faulty, and always passes in a fault-free system. Furthermore, for each unit f_i there is at least one test which is complete for f_i. Any unit other than f_i that can affect the outcome of t_j is said to "invalidate" t_j. As mentioned above, we assume asymmetric invalidation. The following notation from [5] is used:

$f(t_j)$ = the units for which t_j is complete.

$t(f_i)$ = the set of tests which are complete for f_i.

$f(F^i) = t(f_h) \cup \ldots \cup t(f_k)$ for $f_h,\ldots,f_k \in F^i$.
(i.e., the set of tests which are complete for members of F^i).

$T(f_i)$ = the set of tests invalidated by units f_i.

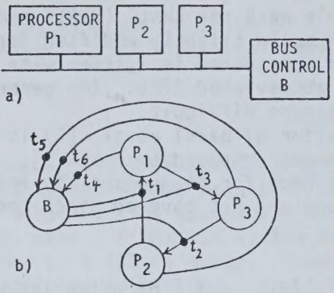
$T(F^i)$ = the set of tests invalidated by fault pattern F^i.

Modeled systems are assumed to be morphic [3] i.e., $T(F^i) = t(f_h) \cup \ldots \cup t(f_k)$ for $f_h,\ldots,f_k \in F^i$.

The relationship between fault patterns and possible test results can be summarized in the generalized fault table or G-array [5]. The G-array has a column for each test, and a row for each fault pattern. Fault patterns are listed in order of increasing number of faults. Entry G^i_j (row F^i, column t_j) is: 0 if t_j will definitely pass, 1 if t_j will definitely fail, X if t_j will give unpredictable results, in the presence of fault pattern F^i. Through use of the morphic property, any row of the G-array can be generated from the first n rows. Thus the first n rows (corresponding to all single faults) constitute a complete description of the system for diagnostic purposes.

The set of pass/fail test results (0's and 1's) obtained by applying all the tests in the system is called a syndrome. A row G^i of the G-array represents all the syndromes that are consistent with fault pattern F^i.

A very useful representation for morphic systems is a directed graph showing the relationships between units, tests, and invalidations in the system. Here we depart slightly from [5] and use a bipartite graph, with unit nodes representing members of F, and test nodes representing members of T. Each test node has incoming edges from all units that can invalidate the test, and one outgoing edge to the unit for which the test is complete. A test node with no incoming edges represents a "hardcore" test--a test that



c)

	B	P_1	P_2	P_3	t_1	t_2	t_3	t_4	t_5	t_6
F(1)	1	0	0	0	X	X	X	1	1	1
	0	1	0	0	1	X	0	X	0	0
	0	0	1	0	0	1	X	0	X	0
	0	0	0	1	X	0	1	0	0	X
F(2)	1	1	0	0	1	X	X	1	1	1
	1	0	1	0	X	1	X	1	1	1
	1	0	0	1	X	X	1	1	1	1
	0	1	1	0	1	1	X	X	X	0
	0	1	0	1	1	X	1	X	0	X
	0	0	1	1	X	1	1	0	X	X
F(3)	1	1	1	0	1	1	X	1	1	1
	1	1	0	1	1	X	1	1	1	1
	1	0	1	1	X	1	1	1	1	1
	0	1	1	1	1	1	1	X	X	X
	1	1	1	1	1	1	1	1	1	1

Fig. 1. Simple system, diagnostic graph and G-array.

is not invalidated by any fault. The diagnostic graph with labeled tests and units contains the same information as the first n rows of the corresponding G-array.

Fig. 1 shows a three-processor system, its (simplified) diagnostic graph, and the associated G-array. To reduce cluttering, arrowheads are left off the invalidation arcs. Arcs with no arrowhead are understood to be directed toward the test node. For more detailed discussions of diagnostic modeling, see [6].

Rather than dealing with the entire system S and its generalized fault table G, we frequently wish to deal only with some subset A of the units in S. For this purpose, we define the diagnostic graph of the subsystem A as the subgraph of S obtained by deleting all unit nodes not in A, and deleting all test nodes not in t(A). Edges which terminate at a deleted node are also deleted. Associated with this subsystem is a generalized fault table $\hat{G}$, which is a subarray of G. The columns of $\hat{G}$ are all the tests that are complete for members of A, and the rows of $\hat{G}$ are all the fault patterns in A.

Finally, we state two important system attributes defined in [3,4]: masking index and closure index. A unit f_a is said to be masked if all tests on that unit are invalidated by faulty units. The masking index of a unit f_a, denoted $m(f_a)$, is the cardinality of the smallest fault pattern that masks f_a. The system masking index m(S) is $\min(m(f_i))$ for all f_i in S. In Fig. 1, m(S) is 1.

A set of units F^i such that $t(F^i) \subseteq T(F^i)$ (that is, every member of the set is masked by the others) is called a closed set. The system closure index c(S) is the cardinality of the smallest closed set in S. In Fig. 1 the smallest closed set is the three processors, so c(S) is 3.

B. Asymmetric Invalidation

In this section we give justification for the use of asymmetric rather than symmetric invalidation in our system model. The asymmetric and symmetric invalidation assumptions differ only in the situation where a faulty unit f_a is involved in a test on another faulty unit f_b. Asymmetric invalidation means that such a test always fails, whereas symmetric invalidation means that any test outcome is possible.

First it must be understood that the "units" in this paper are intended to model fairly complex circuits (memories, microprocessors, etc.) rather than circuits with just a few gates. Consequently, a test which is complete for some unit must actually consist of a sequence of test elements, in which many test responses are checked against expected values.

With this in mind, we consider three common types of test situations in which invalidation occurs. In one case, faulty unit f_a is involved in the test on faulty unit f_b because it is the source of tests or is located somewhere in the signal path running from the test source through the tested unit f_b to the receiver of test responses. For example, faults in a bus could affect tests on any unit attached to the bus. In such situations, the probability of a fault in f_a modifying the test sequence in such a way that all the erroneous responses by f_b are always corrected seems extremely small. Thus a failed test result would be expected.

The second case is when f_a and f_b are identical units, and f_b is tested by applying the same signals to both units and comparing responses. Clearly, the probability of the test passing falsely is the probability of equivalent failures in f_a and f_b. In units that are sufficiently complex, carefully designed and well manufactured, this probability should be insignificant.

The third case is when faulty unit f_a is a unit that receives the test responses from f_b, judges them and reports the pass/fail test result. If this "receiver" is a special-purpose checking circuit, it is possible that a fault could cause some elements of the test sequence to pass falsely, but it is very likely that it will also cause at least one test element to fail, thus producing a "failed" test result. The number of faults causing a false pass could be even further reduced through use of self-checking circuits in the receiver. If the receiving unit is a processor, most faulures that could affect the test results would either cause the test to fail, cause noticeable errors in the message, or result in a "crash" preventing any report at all. A possible exception might be sequencing errors that cause the receiver to prematurely jump to a "test successfully completed" state. Even this unlikely case might be detected if a few simple consistency checks are made, (e.g., How many tests were applied? What were starting and completion times? etc.).

Invalidation situations other than the three considered here may exist, but these three seem to predominate heavily. We conclude that, with some care in the design of units, tests, and test reporting, the probability of a faulty unit passing a test because of invalidation should be negligibly small. In fact, a larger source of error is likely to be tests that falsely pass due to the fact that real tests do not actually cover 100% of possible failures. Thus, in most systems asymmetric invalidation appears to be a reasonable assumption. In the analysis of ultra-reliable systems, however, some of the factors ignored here (such as simultaneous equivalent failures) may become significant, and the more conservative symmetric invalidation assumption might be more appropriate.

II. DIAGNOSABILITY WITH REPAIR

Russell and Kime defind a system to be *t-diagnosable with repair* [3] (t-dwr) if and only if one application of tests allows at least one faulty unit to be identified, provided there are no more than t faults in the system. Thus a t-dwr system with t or fewer faults can be restored to fault-free status in t or fewer test-replace cycles. Stated in terms of the G-array, a system is t-dwr if and only if for $F^1,\dots,F^k \varepsilon F(t)$, $F^1 \cap \dots \cap F^k = \emptyset \Rightarrow G^1 \sqcap \dots \sqcap G^k = \emptyset$. (See [7] for the definition of the cubical operator "$\sqcap$.") In other words, if a system is t-dwr, then all fault patterns in F(t) that produce the same syndrome must have at least one fault in common.

In order to determine conditions for t-dwr, we first examine the characteristics of systems that are *not* t-dwr.

Theorem 1. Let $F^1,\dots F^k$ be any set of fault patterns in F(t) such that $F^1 \cap \dots \cap F^k = \phi$ and $G^1 \sqcap \dots \sqcap G^k \neq \phi$. Let $F^1 \cup \dots \cup F^k = C$, and let $\hat{G}$ be the G-array for the subsystem associated with C. Then:

i) Subsystem C is not t-dwr, and the undiagnosable syndrome $\hat{G}^i \sqcap \ldots \sqcap \hat{G}^k$ is (1,1,...,1), denoted Σ_1.

ii) C is a closed set.

Proof: If $G^i \sqcap \ldots \sqcap G^k \neq \emptyset$, then clearly in the subarray $\hat{G}$, $\hat{G}^i \sqcap \ldots \sqcap \hat{G}^k = 0$, and thus C is not t-dwr. Let t_j be any test which is complete for $f_j \varepsilon C$. By the definition of C, f_j is a member of some fault pattern $F^a \varepsilon \{F^i, \ldots, F^k\}$. Then $\hat{G}^a_j = 1$, because faulty units always fail all tests. Thus for every test t_j on a member of C there is some fault pattern F^a such that $\hat{G}^a_j = 1$, i.e., there is a 1 in every column of the subarray formed by $\hat{G}^i, \ldots, \hat{G}^k$. Since $\hat{G}^i \sqcap \ldots \sqcap \hat{G}^k \neq \emptyset$, we have $\hat{G}^i \sqcap \ldots \sqcap \hat{G}^k = (1,1,\ldots,) = \Sigma_1$. This proves (i).

Since $F^i \sqcap \ldots \sqcap F^k = \emptyset$, there is some pattern F^b such that $f_j \notin F^b$. $G^b_j \neq 1$ because f_j is not faulty. $G^b_j \neq 0$ because $G^a = 1$ and $G^a_j \sqcap G^b \neq \emptyset$. Hence $G^b_j = X$. Thus for every test t_j on a member of C there is a fault pattern F^b such that $G^b_j = X$, i.e., every test on a member of C is invalidated by another member of C. Therefore, C is a closed set. □

Theorem 1 says that in any system which is not t-diagnosable, there must be some closed set C such that when all the tests on members of C fail, diagnosis is impossible. Such a set will be called a t-unreadable set. More formally, a set of units C is t-unreadable if there exist fault patterns $F^i, \ldots, F^k \varepsilon F(t)$ in C such that: $F^i \cap \ldots \cap F^k = \emptyset$, and each of the patterns $F^i, \ldots, F^k$ can produce Σ_1 in subsystem C. A set is t-readable if it is not t-unreadable, and the readability of a set A is the maximum value of t such that A is t-readable.

Corollary 1. A system S is t-dwr if and only if all of its closed subsets are t-readable.

Proof: A proof is in [6]. Just as Corollary 1 reduces the problem of determining diagnosability to an examination of closed sets, the following theorem reduces the task further by showing that we need only consider those closed sets that are strongly connected. A set is strongly connected if there exists a directed path from each node to every other node in the set.

Theorem 2. A System S is t-dwr if and only if all strongly connected closed subsets of S are t-readable.

Proof: A proof is in [6].

A. Analysis and Diagnosis Methods

Theorem 2 shows that the diagnosability with repair of a system is just the minimum of the readabilities of its strongly connected closed sets. Let us consider how one could determine the readability of some closed Set C. As an illustration, we use Fig. 1 which is a closed set.

A fault pattern $F^j \subseteq C$ which can produce Σ_1 in C is referred to as a "blanking pattern." If C is t-unreadable, then by definition, C contains blanking patterns $F^i, \ldots, F^k \varepsilon F(t)$ such that $F^i \cap \ldots \cap F^k = \emptyset$. If C is t-readable, then the intersection of all the blanking patterns in F(t) must contain at least one fault (so that if the Σ_1 syndrome appears, at least one faulty unit can be located). In order to determine diagnosability, then, we could intersect all the blanking patterns in F(1), F(2), F(3), etc., until the intersection becomes empty. If the intersection is empty for the first time at F(x), then readability = x-1. In the example of Fig. 1, the blanking patterns in F(1) and F(2) have non-null intersection (all contain the unit B). The blanking patterns in F(3), however, have no unit in common, so the set is 2-readable. It is shown in [6] that the above approach leads to a simplification of the condition for t-dwr given by Barsi et al. [2] for SMPT systems.

We now have a method for determining the readability of a set. To find the system diagnosability, we must find the minimum readability among all the strongly connected closed sets in the system. Since a strongly connected closed set may itself contain many strongly connected closed subsets, the total number of such sets in a system is often quite large.

In cases where a closed set is contained in another closed set, the readability of the larger set is usaully greater than the readabilities of its subsets. If this property held in general, then it would only be necessary to analyze minimal strongly connected closed sets. Unfortunately, there are exceptions (see Fig. 2). Thus even the closed sets that contain other closed sets must be analyzed. The set with minimum readability in Fig. 1 is $\{P_1,P_2,P_3\}$ which is 1-readable. Hence the system is 1-dwr.

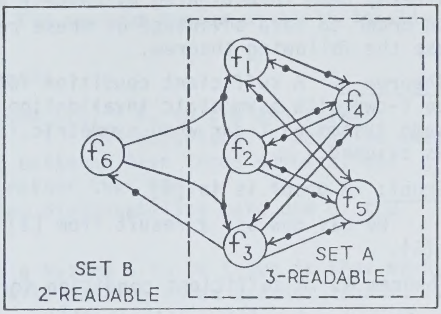


Fig. 2. Example of a set with smaller readability than its subsets.

So far, we have considered only the "analysis" algorithm used to determine the diagnosability of a system. Next, we consider the "diagnosis" algorithm. Given a syndrome, the diagnosis algorithm must identify at least one faulty unit. In most cases the procedure can be very simple. First, we find the set of known good units--units which pass at least one test. Then the tests performed by good units can be used to locate faults. As the set of good units grows larger, more faulty units become identifiable.

This method of fault location breaks down, however, if the Σ_1 syndrome appears in any closed set. In this case, two other simple methods may be tried. First, if any unit in the closed set has a masking index greater than the assumed number of faults (t), then that unit must be faulty. Also, any unit outside the closed set which has both 1 and 0 tests must be a good unit, and the "1" test must have been caused by some faulty unit. Consideration of such faulty tests may allow a fault to be located.

In the worst case, however, the closed set will not yield to these methods, and a "brute force" diagnosis becomes necessary. This consists of partial construction of the G-array for the closed set having the Σ_1 syndrome, as was done in the analysis algorithm. The intersection of all blanking patterns having t or fewer faults is the set which is definitely faulty. Another possibility would be to record all this information during the original diagnosability analysis and then do the fault location via a look-up table.

The "analysis" and "diagnosis" algorithms which were discussed in this section are presented more concisely in [9] along with a few shortcuts that reduce the computation time in most cases.

B. Approximate Methods

The preceding section showed that algorithms for determining dwr and for performing on-line diagnosis can be written. In many cases, however, these algorithms may be much too time-consuming for practical use, For this reason, we now investigate approaches which employ simpler algorithms, but which consequently do not take advantage of a system's maximum possible diagnosability. These methods will, in general, require that the number of faults in the system be kept significantly lower than the actual system diagnosability, which means that fault detection tests will have to be applied more often. The system of Fig. 3 is used for illustration.

One method for estimating diagnosability is to base the estimate on c(S), the size of the smallest closed set in the system. The relationship between c(S) and diagnosability for systems with symmetric invalidation has already been investigated by Russell and Kime [3]. In order to take advantage of these results, we use the following theorem.

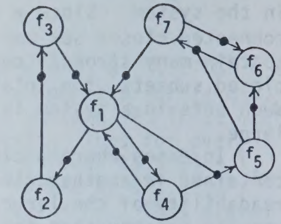


Fig. 3.

Theorem 3. A sufficient condition for a system to be t-dwr with asymmetric invalidation is that the same system be t-dwr when symmetric invalidation is assumed.

Proof: A proof is in [6].

We may now use a result from [3] concerning c(S).

Theorem 4. A sufficient condition for a system S to be t-dwr is $c(S) > \lfloor ((t+2)/2)^2 \rfloor$, or equivalently, $t \leq \lceil 2\sqrt{c(S)} \rceil - 3$.

Proof: The proof is immediate from theorem 3 above and theorem 6 in [3]. □

The existence of the class of systems shown in Fig. 4 indicates that Theorem 4 is the best condition based on closure that can be obtained. (For more detail see [6]). This result is somewhat surprising, since in most cases asymmetric invalidation would be expected to yield greater diagnosabilities than symmetric invalidation. The system of Fig. 3 has a closure index of 7, so according to Theorem 4 its diagnosability is at least $\lceil 2\sqrt{3} \rceil - 3 = 3$.

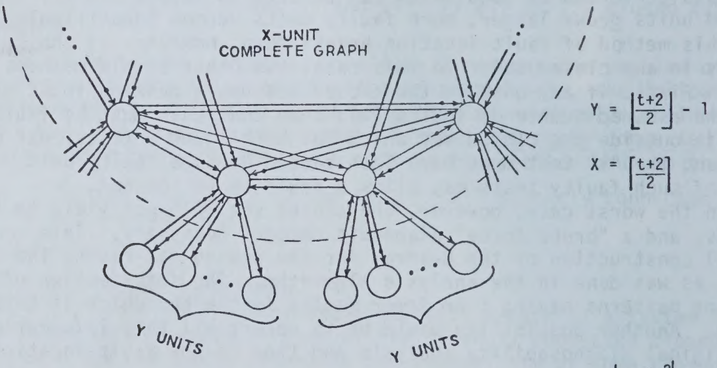

X-UNIT
COMPLETE GRAPH
$Y = \lfloor \frac{t+2}{2} \rfloor - 1$
$X = \lceil \frac{t+2}{2} \rceil$
Y UNITS
Y UNITS

Fig. 4. Example of a system that is not diagnosable for $c(S) = \lfloor (\frac{t+2}{2})^2 \rfloor$.

Estimation of dwr based on system closure makes the analysis task easier, but it does not allow any simplification of the diagnosis algorithm. It is still possible to encounter a Σ_1 syndrome in a closed set, so the original complex diagnosis algorithm is still necessary.

One way to make diagnosis easier is to find a way to keep the number of faults in the system low enough so that the "difficult" case of the Σ_1 syndrome in a closed set never arises.

All that is needed is an analysis algorithm that can determine the size of the smallest blanking pattern in the system. The number could be determined exactly by an exhaustive procedure that would find all closed sets and determine the smallest blanking pattern in each. On the other hand, an estimate of the size of the smallest blanking pattern could be obtained with a great deal less computation through the following theorem.

Theorem 6. A sufficient condition for a system S to be t-dwr is $t < c(S)m(S)/(b(S)+m(S))$ where $b(S)$ is the maximum number of tests invalidated by any unit of S.

Proof: A proof is in [6] and the Appendix.

In Fig. 3 we find $c(S) = 7$, $m(S) = 1$, and $b(S) = 4$, so the approximate method gives diagnosability estimate $t = 1$. In comparison, a detailed analysis shows that the smallest blanking patterns have three numbers, and therefore two faults could be allowed rather than one.

The last two methods for estimating diagnosability make use of the masking index.

Theorem 7. A sufficient condition for a system S to be t-dwr is that each closed set contains at least one unit with masking index greater than t.

In Fig. 3, the maximum masking index is $m(f_1) = 3$, so the diagnosability estimate is $t = 2$.

All of the preceding approaches required either an analysis of every closed set, or computation of system closure index. Unfortunately, for large systems both of these computations may be impractical. Therefore we should consider an even simpler approach, even if it is much less accurate.

Theorem 8. If the number of faults in a system is less than the system masking index $m(S)$, then a unit is faulty if and only if it fails all tests.

The estimate given by this simple method is sometimes very poor. In the case of Fig. 3, $m(S) = 1$ so the diagnosability estimate is 0. We will see further application for this method in the section on diagnosis without repair.

In summary, five different methods for estimating diagnosability with repair, with varying degrees of complexity and accuracy, have been presented:

1. The exact method: t = minimum readability over all strongly connected closed sets. (Theorem 2)
2. Estimate from size of smallest closed set $t = |2\sqrt{c(S)}| - 3$. (Theorem 4)
3. Estimate the size of the smallest blanking pattern by $|F^B| = c(S)m(S)/(b(S)+m(S)))$. Let $t = |F^B| - 1$. (Theorem 6)
4. Prevent blanking by having one unit with masking index > t in each closed set. $t = \min(\max m(f_i) \text{ in } C)$ over all strongly connected closed sets C. (Theorem 7)
5. Estimate from masking index $t = m(S) - 1$. (Theorem 8)

III. DIAGNOSIS WITHOUT REPAIR

A system is t-diagnosable without repair (t-dor), if all faulty units can be located by just one set of test applications, provided that there are no more than t faults present. This is possible only if no two fault patterns in F(t) can cause the same syndrome. We determine necessary and sufficient conditions for t-dor, and consider two types of analysis and on-line diagnosis algorithms.

The necessary condition is exactly the same as that proved by Russell and Kime (Lemma 2, [4]) for systems with symmetric invalidation:

Theorem 9. A necessary condition for a system to be t-dor is $m(S) \geq t$.

The proof is the same as is given in [4]. Briefly, the proof shows that if $m(S) < t$, then there are two fault patterns in F(t) which can cause the same syndrome. One pattern is a masking set F^i with $|F^i| < t$, and the other is F^i along with the unit masked by F^i.

A class of systems that attain the highest possible diagnosability, $t = m(S)$, is given in [2], Theorem 3.

Theorem 10. A sufficient condition for a system S to be t-dor is $m(S) \geq t + 1$.

Proof: A proof is in [6] and the Appendix.

Theorem 10 compares favorably with the sufficient conditions derived by Russell and Kime for systems with symmetric invalidation. Combining Theorems 9 and 10, we find that the diagnosability without repair of any system S is either m(S) or m(S) - 1. This is a rather pleasing result since the masking index is fairly easy to determine. To determine whether the system diagnosability is m(S) or m(S) - 1, a much more complex analysis is required, as indicated by the following necessary and sufficient condition.

Theorem 11. A system S is t-dor if and only if:

i) $m(S) \geq t$

ii) If f_a and f_b are two units in S such that f_a invalidates a test on f_b and f_b invalidates a test on f_a, and if F^A and F^B are two fault patterns in F(A) such that $f_a \epsilon F^A$, $f_b \epsilon F_B$, F^A masks f_b, and F^B masks f_a, then there exists some member of $F^B - F^A$ which is not masked by F^A, or some member of $F^A - F^B$ not masked by F^B.

Proof: A proof is in [6] and the Appendix.

Theorem 2 of Barsi, et al. [2] for SMPT systems is a special case of Theorem 11.

Finding diagnosability by Theorem 11 can be a very complicated process. Furtherfore, if the diagnosability turns out to be m(S), the diagnosis algorithm also becomes difficult. In view of these drawbacks, an attractive alternative is to abandon the t = m(S) cases and always use t = m(S) - 1 (Theorem 10). Then the simple diagnosis algorithm--"repair units that fail all tests"--is always applicable. This approach has merit, not only because of its simplicity, but also because little or none of the potential system diagnosability is sacrificed to attain this simplicity.

V. CONCLUSIONS

System faults can be kept under control by periodic applications of diagnostic tests. Asymmetric test invalidation appears to be a reasonable assumption when the units of the system model are of at least MSI complexity. Compared to symmetric invalidation, the asymmetric assumption leads to at least as high diagnosabilities, and tends to make diagnosis simpler, since every passed test identifies a "good" unit.

Diagnosability with repair of any system can be determined through analysis of all its strongly connected closed sets. A variety of simpler approximate methods can also be used. Methods for preventing the "all 1's" syndrome in a closed set are of particular interest because they allow use of an extremely simple diagnosis algorithm.

Diagnosability without repair can be estimated very simply from the masking index, and a simple diagnosis algorithm is available. Determining whether the actual diagnosability is m(S) or m(S) - 1 is quite complex and may not be worthwhile in many applications.

REFERENCES

[1] P. F. Preparata, G. Metze, and R. T. Chien, "On the connection assignment problem of diagnosable systems," IEEE Trans. Electronic Comput., Vol EC-16, December 1967, pp. 848-854.

[2] F. Barsi, F. Grandoni, and P. Maestrini, "A theory of diagnosability of digital systems," IEEE Trans. Comput., Vol. C-25, June 1976, pp. 585-593.

[3] J. D. Russell and C. R. Kime, "System fault diagnosis: closure and diagnosability with repair," IEEE Trans. Comput., Vol. C-24, November 1975, pp. 1078-1089.

[4] ----, "System fault diagnosis: masking, exposure, and diagnosability without repair," IEEE Trans. Comput., Vol. C-24, December 1975, pp. 1155-1161.

[5] C. R. Kime, "An analysis model for digital system diagnosis," IEEE Trans. Comput., Vol. C-19, November 1970, pp. 1063-1073.

[6] C. S. Holt, and J. E. Smith, "Diagnosis of systems with asymmetric invalidation," Tech. Report ECE-79-18, Dept. of Elec. and Comput. Engr., Univ. of Wisconsin, Madison, WI, 1979.

[7] J. D. Russell, "On the diagnosability of digital systems," Ph.D. dissertation, Univ. of Wisconsin, Madison, WI, July 1973.

ASYMMETRIC TESTING FOR HYBRID FAULT DIAGNOSIS*

SIVANARAYANA MALLELA AND GERALD M. MASSON
Department of Electrical Engineering
The Johns Hopkins University
Baltimore, Maryland 21218

ABSTRACT

We consider the use of asymmetric tests with a certain built-in immunity to invalidation by certain faults for the diagnosis of a class of faults called hybrid faults which are explicitly bounded combinations of permanent and intermittent faults. A general framework is established with which to consider these faults and to justify the possible outcomes which can result when these tests are applied. Necessary and sufficient conditions for diagnosis are given. Tradeoffs between the combinations of intermittent and permanent faults within a system are considered such that diagnosis is correct.

---

* This research was supported by NASA Grant NSG 1442.

INTRODUCTION

System level fault diagnosis continues to grow in importance now that the validation of fault tolerant computing has emerged as one of the more prominent research problems in computer engineering. In this paper we address systems level diagnosis for hybrid fault situations. Necessary and sufficient conditions are established relative to interrelationships among faults and special types of tests which have a certain built-in immunity to invalidation such that a special type diagnosis can be achieved. The tests involved will be assumed to have been generated for the extreme hybrid fault case involving all permanent faults. Hence, it is of interest to consider the possible trade-off involved relative to the extent that intermittent faults can be involved in a fault situation when it is known to what extent this all permanent fault case can be handled. Along these lines it will be seen that only one type of asymmetric test must really be considered.

Finally, the results presented in this paper are stated without proof because of page limitations. However, a complete exposition of hybrid fault diagnosis in digital systems can be found in S. Mallela's Ph.D. thesis: "Hybrid Faults in Computing Systems: Diagnosis without Repair," Department of Electrical Engineering Department, The Johns Hopkins University, December, 1979.

Preliminaries

We define a system, S , to be an assembly of digital components which is partitionable into subsystems. Accordingly, we will represent a system as the set $S = \{S_1, S_2, \ldots, S_i, \ldots, S_n\}$, where each element, say, S_i refers to a subsystem.

Each subsystem of S has its own set of inputs and outputs. The functional behavior of a subsystem is characterized by its output response to the set of the different stimuli which can be applied at its inputs. Regardless of the level of system under consideration, the physical components comprising the subsystem are subject to various failures which cause subsystem malfunctions in the sense that this functional behavior is incorrect. Fault diagnosis can then be described, at this point, as the unambiguous identification or location of malfunctioning subsystems within the entire system.

For any subsystem, say, S_i , we will refer to its correct functional behavior as the *fault-free functional behavior* and denote it as X_{i0} . In the presence of physical component failures, the subsystem S_i malfunctions in that it exhibits a functional behavior which is different from X_{i0} . Thus, for each subsystem, say, S_i we will also refer to the *set of faulty functional behaviors*, which we will denote as $\{X_{i1}, X_{i2}, \ldots , X_{im_i}\}$, where each X_{ik}, $k=1, \ldots, m_i$, is a unique functional behavior which the malfunctioning subsystem can exhibit. We can then denote the complete set of functional behaviors (fault-free and faulty) of the subsystem S_i as $\{X_{i0}, X_{i1}, \ldots, X_{im_i}\}$.

A system will be said to malfunction if any of its subsystems malfunctions. A subsystem malfunctions as a result of a physical component failure. We will use the term *fault* to refer to the occurrence of any such failure which results in the malfunctioning of a subsystem. Hence, for a system S which is partitioned into the set of subsystems $\{S_1, \ldots, S_n\}$, we will refer to its *fault set*, denoted by $F = \{f_1, \ldots , f_n\}$. Recall that $\{X_{i1}, \ldots, X_{im_i}\}$ is the set of faulty functional behaviors of S_i . Then, in F we will let $f = \{f_{i1}, \ldots, f_{im_i}\}$ be the set of physical component failures which cause S_i to exhibit one of these faulty functional behaviors in the sense that f_{i1} is the collection of physical component failures of S_i which result in the functional behavior $X_{i1}, \ldots,$ and f_{im_i} is the collection of physical component failures of S_i which result in the functional behavior X_{im_i} .

We define a set of stimuli applied to the inputs of system to be a *test* if there exists some malfunction(s) of the system such that the response of the malfunctioning system to this set of stimuli is observably different from that of the fault-free system. With a given system S , we associate a set of tests, $T = (t_1,\ldots,t_j,\ldots,t_v)$ for the faults $\{f_1,f_2,\ldots,f_n\}$ in the system where each $t_j \varepsilon T$ corresponds to some specified set of input stimuli. Each test $t_j \varepsilon T$ detects system malfunctions by detecting malfunctions in certain subsystem(s) of the system. Furthermore, since in general most inputs and outputs of subsystems within the system are not accessible for direct stimulation and observation of responses, the application of a test $t_j \varepsilon T$ usually involves the use of various subsystems within the system.

The application of a test $t_j \varepsilon T$ is assumed to be accompanied by the comparison of the observable response of the system to t_j with the known corresponding response of the fault-free system. A system is said to *pass* the test when the observable response of the system is identical to that of the fault-free system; otherwise, it is said to *fail* the test. We associate an *outcome* a_j to t_j , specified by a binary value in the following way: $a_j = 0$ if t_j is failed by the system. It is the collection of outcomes of tests in T which, upon analysis, is the basis of fault diagnosis in the system. A complete set of tests outcomes, that is an outcome a_j for each $t_j \varepsilon T$, is called a *syndrome*. Since a_j is either 0 or 1 , it follows that a syndrome, say, S_i is a binary v-typle, where $v = |T|$. We define a syndrome, say, S_x to be a *subsyndrome* of a syndrome, say, S_y if $a_j= 1$ in S_x implies that $a_j=1$ in S_y .

There will usually be a delay between the application of the first stimuli corresponding to t_j and the specification of a_j . This time period will be referred to in general as the *duration of the test* or, in this case, the duration of t_j . The time interval between the application of the first test and the specification of the outcome of the last test to be applied will be referred to as the *testing period*. We define a fault, say, f_i to be *present* in the system if the probability that the subsystem S_i exhibits the functional behavior X_{i0} for the entire testing period, is less than 1 . Otherwise, we define the fault f_i to be not present. Along similar lines, we refer to a set of faults $F_i \subseteq F$ is being present in the system if each fault $f_i \varepsilon F_i$ is present in the system.

Note that the functional behavior of a system is characterized by the functional behaviors of subsystems within the system. When a set of faults $F_i \subseteq F$ is present in the system, each subsystem S_i which corresponds to a fault $f_i \varepsilon F_i$ can exhibit any one of m_i different faulty functional behaviors. Accordingly, for each set of faults $F_i \subseteq F$, we will define the set of system functional behaviors, which we will denote by $Z(F_i)$. Each element in $Z(F_i)$ describes the functional behavior exhibited by the system when each subsystem S_i , which corresponds to a fault $f_i \varepsilon F_i$, exhibits a specific functional behavior X_{ik} , $k \neq 0$. Clearly, the set $Z(F_i)$ contains $(\prod_{f_i \varepsilon F_i} m_i)$ elements.

Modes and Types of Faults

Given that f_i is present in the system, we will say that f_i is active at some given time if the subsystem S_i exhibits a functional behavior X_{ik}, $k \neq 0$ at that time. Otherwise, we will refer to the fault as being inactive. There are three different modes of activity/inactivity that f_i can exhibit.

(i) Uniformly inactive mode: We define a fault, say, f_i to be uniformly inactive during the time interval ∇ if for the entire interval ∇ , the subsystem S_i exhibits the functional behavior X_{i0} .

(ii) Uniformly active mode: We define a fault, say, f_i to be uniformly active during the time interval ∇ if for the entire interval ∇ , the subsystem S_i exhibits a functional behavior X_{ik} , where $k \neq 0$.

(iii) Nonuniform mode: We define a fault, say, f_i to be nonuniform during the time interval ∇ if there are at least two disjoint subintervals of time, say, δ_1 and δ_2 in ∇ such that S_i exhibits a functional behavior X_{ik} during δ_1 and a functional behavior $X_{i\ell}$ during δ_2 , where $k \neq \ell$.

Based upon the notions of modes of activity and inactivity and together with our concept of a testing period, we can now classify a fault as being one of two types:

(i) Permanent fault: We define a fault, say, f_i , which is present in the system, to be a permanent fault if f_i can only be uniformly active during any given subinterval of time within the testing period. In such a case, clearly, the subsystem S_i would exhibit the same functional behavior, X_{ik}, $k \neq 0$, for the entire testing period.

(ii) Intermittent faults: We define a fault, say, f_i , which is present in the system, to be an intermittent fault if f_i cannot only be uniformly active during any given subinterval of the testing period (which is the case for permanent faults) but if f_i can also be uniformly inactive or nonuniform during any given subinterval of the testing period.

In this paper, we will consider the most general of all fault situations which we will call hybrid faults and which can be described as consisting explicitly bounded combinations of permanent and intermittent faults.

Detection Components and Invalidation Components

The collection of test outcomes, upon analysis, forms the basis of fault diagnosis. However, in order that syndromes which result from test set applications can be analyzed for the purpose of identifying malfunctioning subsystems, the possible outcomes for any test $t_j \epsilon T$ must be known for different sets of faults $F_i \subseteq F$ in the system. Moreover, as we are considering the potential presence of combinations of permanent faults and intermittent faults in the system, the outcome possibilities of a test $t_j \epsilon T$ must be known for all fault situations wherein each fault $f_i \epsilon F_i$ is either a permanent fault or an intermittent fault. Hence, we will provide the test outcome specification as follows: first, for each test $t_j \epsilon T$ and each set of faults $F_i \subseteq F$, we will specify the test outcome a_j for the case where F_i is the set of faults in the system, and all the faults within F_i are permanent faults; second, we will utilize this test outcome specification for permanent fault situations to specify the test outcome a_j for each fault situation wherein some or all of the faults within F_i are intermittent.

With a test $t_j \epsilon T$ and a set of faults $F_i \subseteq F$, we associate the set of outcomes $a_j(F_i)$ which is the set of possible values of a_j when F_i is the set of faults in the system, and all the faults within F_i are permanent faults. When F_i is the set of faults in the system and all the faults in F_i are permanent faults, the functional behavior of the system is described by some element of $Z(F_i)$. Relative to the set $Z(F_i)$, a test t_j must fall into one of the following three categories:

(i) $a_j(F_i) = \{0\}$;

(ii) $a_j(F_i) = \{1\}$;

(iii) $a_j(F_i) = \{0,1\}$.

In order to diagnose permanent fault situations on the basis of syndromes, it is necessary to know $a_j(F_i)$ for all tests $t_j \epsilon T$ and all sets of faults $F_i \subseteq F$. However, because, in general, a system S could have a large number of different sets of faults associated with it, it is impractical to provide an individual specification of $a_j(F_i)$ for each pair $t_j \epsilon T$ and $F_i \subseteq F$. Hence, in the following, we will establish two interrelationships between tests and faults which will be referred to as detection components and invalidation components of tests. As will be seen, for a test $t_j \epsilon T$, the detection components and invalidation components are sets of faults which are usually much more tractable when compared to the number of subsets $F_i \subseteq F$. We will then use these more tractable test-fault interrelationships to determine $a_j(F_i)$ for all pairs $t_j \epsilon T$ and $F_i \subseteq F$.

We define a set of faults $F_i \subseteq F$ to be a detection component of a test $t_j \epsilon T$ if:

(i) $a_j(F_i) = \{1\}$;

(ii) there exists no subset $F'_i \subset F_i$ such that $a_j(F'_i) = \{1\}$.

We next make the following:

Assumption 1: For a given system S , the partition into subsystems is such that for the resulting fault set F and the test set T , we have:

(i) there is at least one detection component for each test $t_j \epsilon T$;

(ii) there exists no subset $F'_i \subset F_i$ such that $a_j(F'_i) = \{1\}$.

With each set of faults $F_i \subseteq F$ we associate the set of detection tests, which we will denote by $DT(F_i)$. This set consists of all tests $t_j \epsilon T$ such that a detection component of t_j is contained within F_i . For each fault $f_i \epsilon F$, it follows that $DT(\{f_i\})$ is the set of all the tests in T for which $\{f_i\}$ is a detection component. Since on the basis of Assumption 1, no detection component of a test contains more than one fault, it follows that for any set of faults $F_i \subseteq F$,

$$DT(F_i) = \bigcup_{f_i \varepsilon F_i} DT(\{f_i\}) \ . \qquad (1)$$

We can now develop our second test-fault interrelationship. Recall that the application of a test $t_j \varepsilon T$ in general requires the use of various subsystems within the system. We will refer to an application of a test $t_j \varepsilon T$ as being valid if all of the operations associated with the test application are performed correctly. We will refer to the application of t_j as being invalid if any such operation is incorrect.

We can now define a set of faults $F_i \subseteq F$ to be an invalidation component of a test $t_j \varepsilon T$ if:

(i) $t_j \not\varepsilon DT(F_i)$;

(ii) there exists at least one system functional behavior that is described by an element of $Z(F_i)$ for which the application of t_j is invalid;

(iii) for all $F_i' \subset F_i$, the application of t_j is definitely valid if the system exhibits any functional behavior that is described by an element of $Z(F_i')$.

If should be pointed out that if a subsystem S_i is used in the application of a test $t_j \varepsilon T$, then it does not necessarily follow that f_i is contained in some invalidation component of t_j .

We next make the following:

Assumption 2: For a given system, S , the partition into subsystems is such that for the resulting fault set F and the test set T , we have that each invalidation component of a test $t_j \varepsilon T$ contains exactly one fault.

With each set of faults $F_i \subseteq F$, we associate the set of potentially invalid tests, which we denote by $IT(F_i)$, which consists of all the tests $t_j \varepsilon T$ such that an invalidation component of t_j is contained within F_i . For each fault $f_i \varepsilon F$, it follows that $IT(\{f_i\})$ is the set of all the tests $t_j \varepsilon T$ for which $\{f_i\}$ is an invalidation component. Since on the basis of Assumption 2, no invalidation component of a test contains more than one fault, it follows that for any set of faults $F_i \subseteq F$,

$$IT(F_i) = \bigcup_{f_i \varepsilon F_i} IT(\{f_i\}) \ . \qquad (2)$$

Test Outcome Specification for Permanent Fault Situations

We will now provide the specification of test outcomes for permanent fault situations by making use of the detection and invalidation components of tests. However, we will first give the following assumption which is implicit in all of the test-fault interrelationship models available in the published literature.

Assumption 3: If a test $t_j \varepsilon IT(\{f_i\})$ for some fault $f_i \varepsilon F$, then there exists at least one X_{ik}, $k \neq 0$, under which S_i correctly performs the required operations associated with the application of t_j .

Suppose now that $F_i \subseteq F$ is the set of faults in the system and all the faults in F_i are permanent faults. For a test $t_j \varepsilon T$, we will specify $a_j(F_i)$ based on whether or not $t_j \varepsilon IT(F_i)$, and whether or not $t_j \varepsilon DT(F_i)$.

The following case is straightforward.

Case I: $t_j \not\varepsilon IT(F_i)$.

(i) $t_j \not\varepsilon DT(F_i)$: $a_j(F_i) = \{0\}$.

(ii) $t_j \varepsilon DT(F_i)$: $a_j(F_i) = \{1\}$.

The next case is somewhat more involved.

Case II: $t_j \varepsilon IT(F_i)$.
In such a case, an application of t_j is potentially invalid in the presence of the set of permanent faults F_i. This implies that there is at least one system functional behavior that is described by an element of $Z(F_i)$ for which the application of t_j is invalid. On the basis of Assumption 3, however, for every fault $f_i \varepsilon F_i$, where $t_j \varepsilon IT(\{f_i\})$, there exists at least functional behavior $X_{ik}, k \neq 0$, under which S_i correctly performs the required operations associated with the application of t_j .

(1) Suppose that the application of t_j is valid in the presence of the set of permanent faults F_i . Then $a_j=0$ if $t_j \not\varepsilon DT(F_i)$, and $a_j=1$ if $t_j \varepsilon DT(F_i)$.

(2) Suppose now that the application of t_j is invalid in the presence of the set of permanent faults F_i . Then, in general,

a_j can be 0 even though $t_j \varepsilon DT(F_i)$, and

a_j can be 1 even though $t_j \not\varepsilon DT(F_i)$.

Tests for which this is the case will be referred to as symmetric tests. However, there exists tests for which only one of the above two kinds of incorrect outcome is possible. We will refer to these as asymmetric tests. We will refer to such a test as a 1-asymmetric test if

a_j can be 1 even if $t_j \not\varepsilon DT(F_i)$, but

a_j cannot be 0 if $t_j \varepsilon DT(F_i)$.

We will refer to such a test as a 0-asymmetric test if

a_j can be 0 even if $t_j \varepsilon DT(F_i)$, but

a_j cannot be 1 if $t_j \not\varepsilon DT(F_i)$.

With the above, the specification of $a_j(F_i)$ can be summarized in the following table.

	Symmetric Tests	1-asymmetric Tests	0-asymmetric Tests
$t_j \not\varepsilon IT(F_i)$ $t_j \not\varepsilon DT(F_i)$	$\{0\}$	$\{0\}$	$\{0\}$
$t_j \not\varepsilon IT(F_i)$ $t_j \varepsilon DT(F_i)$	$\{1\}$	$\{1\}$	$\{1\}$
$t_j \varepsilon IT(F_i)$ $t_j \not\varepsilon DT(F_i)$	$\{0,1\}$	$\{0,1\}$	$\{0\}$
$t_j \varepsilon IT(F_i)$ $t_j \varepsilon DT(F_i)$	$\{0,1\}$	$\{1\}$	$\{0,1\}$

Table 1

In the following, we will take advantage of:

Assumption 4: Each test $t_j \varepsilon T$ is either a symmetric test, a 1-asymmetric test, or a 0-asymmetric test.

Given set of tests $T_i \subseteq T$ could, in general, contain tests of each of the 3 types discussed above, we will use $(T_i)_s$, $(T_i)_{1a}$, and $(T_i)_{0a}$ to denote respectively the set of all symmetric, all 1-asymmetric, and all 0-asymmetric tests contained within T_i .

Specification of Test Outcomes for Hybrid Fault Situations

In this section, we will provide the specification of test outcomes for hybrid fault situations. We begin with the following two assumptions.

Assumption 5: If $t_j \not\in IT(\{f_i\})$ and S_i is employed in the application of t_j , then S_i correctly performs the required operations associated with the application of t_j even when f_i is an intermittent fault.

Assumption 6: When the application of an asymmetric test t_j is invalid, regardless of whether the invalid application is caused by permanent faults or intermittent faults,

(i) $a_j=1$ for a 1-asymmetric test if at least one permanent fault f_i, where $t_j \varepsilon DT(\{f_i\})$, is present in the system, and
(ii) $a_j=0$ for a 0-asymmetric test if no fault f_i is present in the system, where $t_j \varepsilon DT(\{f_i\})$.

Suppose now that $F_i \subseteq F$ is the set of faults in the system where each fault $f_i \varepsilon F_i$ is either a permanent fault or an intermittent fault. Let $F^p_i \subseteq F_i$ refer to the set of all the permanent faults within F_i . For a test $t_j \varepsilon T$, we will now provide a specification of a_j for hybrid fault situation. This specification is given in Table 2 together with a brief justification.

	$a_j(F_i)=\{0\}$	$a_j(F_i)=\{1\}$	$a_j(F_i)=\{0,1\}$
$t_j \not\in IT(F_i)$ $t_j \not\in DT(F_i)$	$a_j=0$ Assumption 5		
$t_j \not\in IT(F_i)$ $t_j \varepsilon DT(F_i)$		$a_j=1$ if $t_j \varepsilon DT(F^p_i)$; $a_j=0$ or 1 otherwise. Assumption 5 Permanent faults can only be uniformly active, while intermittent faults can also be uniformly inactive.	
$t_j \varepsilon IT(F_i)$ $t_j \not\in DT(F_i)$	$a_j=0$ Assumption 6		$a_j=0$ or 1 intermittent faults can be uniformly active
$t_j \varepsilon IT(F_i)$ $t_j \varepsilon DT(F_i)$		$a_j=1$ if $t_j \varepsilon DT(F^p_i)$; $a_j=0$ or 1 otherwise. Assumption 6 Permanent faults can only be uniformly active, while intermittent faults can also be uniformly inactive.	$a_j=0$ or 1 intermittent faults can be uniformly active

Table 2

Allowable Sets of Faults

Difficulties arise in the identification of faults in a system because a given hybrid fault situation can potentially produce a great variety of syndromes. Accordingly, it is common practice in diagnosable systems research to alleviate the extent of the difficulties somewhat by restricting the fault situations which must be handled. We will refer to an allowable fault situation to generally describe those hybrid fault situations which fall into this class.

To determine whether a fault situation involving a set of faults $F_i \subseteq F$ is allowable, we need to first define a measure over sets of faults. Accordingly with each set of faults $F_i \subseteq F$, we associate a real-valued measure, denoted by $m(F_i)$, which is such that:

(i) $m(F_i) > 0$ for all non-empty sets of faults $F_i \subseteq F$, and

(ii) for any set of faults $F_i \subseteq F$, $m(F_i) = \sum_{f_i \varepsilon F_i} m(\{f_i\})$.

A well-accepted measure which has been used predominantly in the literature is the cardinality measure, where the description of a fault situation is in terms of the number of faults in the given fault situation. [1]-[4] Maheshwari and Hakimi [5] have also introduced a measure which is probabilistic.

Now, with the concept of the real-valued measure for allowable hybrid fault situations we will specify an upper bound on the combined measure of the faults (either permanent or intermittent) which can be present in the system and designate this value as K_h. We will also specify an upper bound on the combined measure of the faults which can be intermittent within the set of faults in the system and designate this value as K_{hi}. We will refer to such hybrid fault situations as K_h/K_m-fault situations. We will refer to the capability of performing diagnosis for K_h/K_{hi}-fault situations as K_h/K_{hi}-fault diagnosability.

To describe syndromes produced by allowable fault situations, consider an allowable set of faults F_k for which, clearly, $m(F_k) \leq K_h$. We will denote as the syndrome set $S_h(F_k)$ the set of all syndromes that can possibly be produced by allowable fault situations involving F_k (that is, fault situations where all the faults of some $F_k' \subseteq F_k$, where $m(F_k') \leq K_{hi}$, are intermittent faults, and all the faults of $(F_k - F_k')$ are permanent faults). We will then denote as $S_p(F_k)$ the set of all syndromes that can possibly be produced when all the faults in F_k are permanent faults. It should be celar that a syndrome $s_i \varepsilon S_p(F_k)$ if and only if each test outcome a_j has a value in s_i which is contained in $a_j(F_k)$. Now, since $S_h(F_k)$ must also include the syndromes that can be produced by the allowable fault situation wherein all the faults in F_k are permanent, $S_p(F_k) \subseteq S_h(F_k)$. If any syndrome $s_i \varepsilon S_p(F_j)$ for some allowable set of faults F_j, then we will say that s_i is a permanent fault compatible syndrome (pf-compatible syndrome); otherwise, we will say that s_i is pf-incompatible.

Quality of Diagnosis

The quality of diagnosis which we will be mainly concerned with in this thesis relative to the faults identified on the basis of a given syndrome will be correct diagnosis. With correctness of the diagnosis as a requirement, it will not, in general, be possible to avoid incomplete diagnosis for hybrid fault situations. Our goal then can now be stated as establishing necessary and sufficient conditions on the test-fault interrelationships of diagnosable systems so that correct, but sometimes incomplete K_h/K_{hi}-fault diagnosability is realized, where the incompleteness is solely a consequence of the possible intermittent nature of the faults. We will permit this identification of fault situations to be based on only pf-compatible syndromes. We will henceforth refer to this type of diagnosability as K_h/K_{hi}-fault diagnosability without repair.

Characterization of K_h/K_{hi}-Fault Diagnosability Without Repair

A system s is K_h/K_{hi}-fault diagnosable without repair if and only if the following two conditions are satisfied:

(i) For any two sets of faults, F_i and F_j, where $m(F_i) \leq K_h$, $F_i \neq F_j$, we have that $S_p(F_i) \cap S_p(F_j) = \phi$

(ii) For any allowable fault situation involving a set of faults F_i, where $m(F_i) \leq K_h$, any pf-compatible syndrome s_i that can be produced after a test set application must be such that $s_i \varepsilon S_p(F_j)$ for some $F_j \subseteq F_i$.

The following theorem states the necessary and sufficient conditions on the test-fault interrelationships for K_h/K_{hi}-fault diagnosability without repair when the test set T contains only asymmetric tests.

Theorem 1: A system S is K_h/K_{hi}-fault diagnosable without repair if and only if for any set of faults $F_i \subseteq F$, where $m(F_i) \leq K_h$, at least one of the following two conditions holds relative to any other set of faults $F_j \subseteq F$, where $m(F_j) \leq K_h$, and $F_i \cap F_j = \phi$:

a: For any subset $F_{j1} \subsetneq F_j$, where $m(F_{j1}) \leq K_h - m(F_i)$, at least one of the following conditions, a1 or a2, is satisfied.

a1: $[DT(F_i) - DT(F_i) \cap DT(F_j)]_{1A} \not\subseteq IT(F_j)$;

a2: $[DT(F_i) - DT(F_i) \cap DT(F_j)]_{0A} \not\subseteq IT(F_i \cup F_{j1})$;

b: For every $F_{j1} \subseteq F_j$ which does not satisfy Condition a above, there exists a set $F_j^* \subseteq (F_j - F_{j1})$, such that $m(F_j^*) > K_{hi}$, where F_j^* is the set of all the faults $f_i \varepsilon (F_j - F_{j1})$ for which at least one of the following conditions b1 or b2 holds relative to F_i and F_{j1}:

b1: $[DT(\{f_i\}) - DT(\{f_i\}) \cap DT(F_i \cup F_{j1})]_{1A} \not\subseteq IT(F_i \cup F_{j1})$;

b2: $[DT(\{f_i\}) - DT(\{f_i\}) \cap DT(F_i \cup F_{j1})]_{0A} \not\subseteq IT(F_j)$.

The proof of this theorm is omitted because of its length.

Two special cases of K_h/K_{hi}-fault diagnosability without repair are of particular interest: namely, the case where $K_{hi}=0$ and the case where $K_{hi}=K_h$. If $K_{hi}=0$, then hybrid fault diagnosability without repair reduces to permanent fault diagnosability without repair. We will refer to this as $K_p/0$-fault diagnosability without repair. If $K_h=K_{hi}$, then for any set of faults F_i in the system, where $m(F_i) \leq K_h$, regardless of whether some or all of the faults in F_i are intermittent, the fault situation would be allowable. In such a case, for hybrid fault diagnosability without repair, the restrictions on the intermittent faults which can be present in the system are minimal in the sense that we only require that $m(F_i) \leq K_h$ for any set of faults F_i which can be present in the system. We will refer to this as K_i/K_i-fault diagnosability without repair. The characterizations for the above special cases of K_i/K_{hi}-fault diagnosability without repair can be obtained directly from Theorem 1 by making the substitutions for K_h and K_{hi}.

Note that from Assumption 1 it follows that tests $t_j \varepsilon T$ have been generated for the detection of permanent faults. Therefore, the selection of the tests in T is influenced by the desired permanent fault diagnosing capability. Now, in order to determine if it is worthwhile to also consider the issue of hybrid fault diagnosis in the selection of tests $t_j \varepsilon T$, we must know the guaranteed minimum hybrid fault diagnosing capability in a system with a known permanent fault diagnosing capability. In Table 3, for certain classes of systems which are $K_p/0$-fault diagnosable without repair, we present the guaranteed minimum value for K_i/K_i-fault diagnosability without repair. It should be pointed out that in Table 3, K_p refers to the largest value for which the system is $K_p/0$-fault diagnosable without repair. Furthermore, it is assumed that $M \le K_{p/2}$ since it can be shown that any system which is $K_p/0$-fault diagnosable without repair is guaranteed to be $(\frac{K_p}{2})/(\frac{K_p}{2})$-fault diagnosable without repair.

	Tests could contain more than one detection component	Each test contains only one detection component
All the tests in T are 1-asymmetric tests	$K_p - M$, where $M = \max_{f_i \varepsilon F}(m(\{f_i\}))$.	$K_p - M$, where $M = \max_{f_i \varepsilon F}(m(\{f_i\})$
All the tests in T are 0-asymmetric tests	$\frac{K_p}{2}$	K_p

Table 3

As can be seen from Table 3, three of the four classes of systems considered are such that a system which is $K_p/0$-fault diagnosable without repair is "almost" K_p/K_p-fault diagnosable without repair. Therefore, for such systems it is not necessary for all practical purposes to consider separately the issue of hybrid fault diagnosis in the initial system design. This, however, is not the case if for the system all the tests in T are 0-asymmetric and the tests contain more than one detection component. Indeed, for such a system it would even be worthwhile to consider an intermediate value of K_h , where $K_i < K_h \le K_p$, because of the possible trade-off involved between K_h and K_{hi} .

We conclude this paper with a Theorem which concerns the minimum value of K_{hi} for guaranteed K_h/K_{hi}-fault diagnosability without repair in the above systems.

Theorem 2: Consider a system S where all the tests in T are 0-asymmetric tests, and some tests contain more than one detection component. If the system S is $K_p/0$-fault diagnosable without repair, then for a given K_h , where $\frac{K_p}{2} < K_h \le K_p$, it is also K_h/K_{hi}-fault diagnosable without repair for all values of K_{hi} such that

$$K_{hi} \le K_p - K_h \; .$$

REFERENCES

[1] F. P. Preparata, G. Metze, and R. T. Chien, "On the connection assignment problem of diagnosable systems," IEEE Trans. Electron. Comput., vol. EC-16, pp. 848-854, Dec. 1967.

[2] S. L. Hakimi and A. T. Amin, "Characterization of the connection assignment of diagnosable systems," IEEE Trans. Comput., vol. C-23, pp. 86-88, Jan. 1974.

[3] J. D. Russell and C. R. Kime, "System fault diagnosis: Masking, exposure and diagnosability without repair," IEEE Trans. Comput., vol. C-24, pp. 1156-1161, Dec. 1975.

[4] S. Mallela and G. M. Masson, "Diagnosable systems for intermittent faults, IEEE Trans. Comput., vol. C-27, pp. 560-366, June 1978.

[5] S. N. Maheshwari and S. L. Hakimi, "On models for diagnosable system and probabilistic fault diagnosis," IEEE Trans. Comput., vol. C-25, pp. 228-236, March 1976.

MODIFICATION OF ACAR-GHAUSI SENSITIVITY MEASURE WHICH INCLUDES THE NON-IDEAL EFFECTS OF THE OP. AMPS

J. MULAWKA
School of Engineering, Oakland University, Rochester, MI 48063

Abstract. A modification procedure for Acar-Ghausi statistical sensitivity measure is presented which includes the effects of nonidealness of the op. amps. Calculation of sensitivity is based on a suitable decomposition of the denominator of the transfer function. The measure takes into account tolerances and correlation of the op. amps and can be quickly calculated for different realizations of a circuit admittances.

I. INTRODUCTION

The sensitivity of a circuit is an important practical consideration in the design of active networks. One of the well-known multiparameter sensitivity measures was reported by Rosenblum-Ghausi [1]. There were different formulations of this measure [2-4] which realistically account for passive components tolerances and correlations. In [1] the measure is generalized for high-Q analysis to account for the error in realization due to the operational amplifier (op. amp.) pole. This formulation, however, does not include tolerances and correlations of the op. amps. The sensitivity measure to be practical, all parameter variations must be included. We are also interested in a formulation which would be suitable for the case when coefficients of the denominator of the transfer function are not known a priori. It can be in a general form such as a function of network elements. Thus, the measure can be used in a generation of new low sensitivity active filters via admittance selections [5].

Due to op. amps. poles the degree of the denominator is higher than two. To simplify calculations some decomposition of polynomial is required. In this way we need to calculate integrals only once. The decomposition reported in [4] is not adequate for our purpose since it deals with sensitivities of Q-factor and ω_o. But in our method we are interested in sensitivities of coefficients in the polynomial forms associated with first and second power of variable s. The last postulate makes possible calculation of derivaties in an easy way.

II. DECOMPOSITION OF THE DENOMINATOR

The denominator D of a transfer function may be expressed in the following form

$$D = s^2 U_o \left(1+\frac{n_2}{A}\right) + sW_o\left(1 + \frac{n_1}{A}\right) + \left(1 + \frac{n_o}{A}\right) \tag{1}$$

where

$$\frac{n_i}{A} = \sum_{j=1}^{l} \frac{n_{ij}}{A_j}$$

l - number of op. amps.

n_{ij} - some real numbers

U_o, W_o - parameters of second order denominator which are equivalent

$$U_o = \frac{1}{\omega_o^2}, \quad W_o = \frac{1}{Q\omega_o} \tag{2}$$

Next, we make use of the one-pole op. amp. model given by

$$\frac{1}{A_i} = \frac{1}{A_{oi}} + \frac{s}{A_{oi}\omega_{1i}} = BO_i + sBG_i \tag{3}$$

After substituting (3) into (1), we get

$$D = s^3 U_o \frac{n_2}{A_o\omega_1} + s^2 U_o \left(1 + \frac{n_2}{A_o} - \frac{n_o}{A_o} + \frac{W_o}{U_o}\frac{n_i}{A_o\omega_1}\right) + \tag{4}$$

$$+ sW_o\left(1 + \frac{n_1}{A_o} - \frac{n_o}{A_o} + \frac{1}{W_o}\,\frac{n_o}{A_o\omega 1}\right) + 1$$

Let us consider the following decomposition of (4):

$$D = (s\tau+1) \quad (s^2U+sW+1) \tag{5}$$

where $U = U_o(1+\Delta U)$

$W = W_o(1+\Delta W)$

Comparing (4) with (5) we arrive at the equation set

$$U_o \frac{n_2}{A_o\omega_1} = \tau U_o[1+\Delta U]$$

$$U_o\left(1 + \frac{n_2}{A_o} - \frac{n_o}{A_o} + \frac{W_o}{U_o}\,\frac{n_1}{A_o\omega_1}\right) = U_o(1+\Delta U) + \tau W_o(1+\Delta W) \tag{6}$$

$$W_o\left(1 + \frac{n_1}{A_o} - \frac{n_o}{A_o} + \frac{1}{W_o}\,\frac{n_o}{A_o\omega_1}\right) = W_o(1+\Delta W) + \tau$$

It is easy to check that for high value of Q-factor

$$n_1 \gg n_2 \tag{7}$$

$$n_1 \gg n_o$$

Taking into account (7) and neglecting higher powers and cross products of ΔU and ΔW we arrive at the solution.

$$\Delta U \cong \frac{n_2}{A_o} - \frac{n_o}{A_o} + \frac{W_o}{U_o}\,\frac{n_1}{A_o\omega_1} \tag{8}$$

$$\Delta W \cong \frac{n_1}{A_o} + \frac{1}{W_o}\left(\frac{n_o}{A_o\omega_1} - \frac{n_2}{A_o\omega_1}\right) \tag{9}$$

III. SENSITIVITY MEASURE FORMULATION

For high Q-factors, sensitivity due to the real pole can be neglected. Thus, we are interested in the sensitivity of the polynomial

$$D = s^2 U + sW + 1 \tag{10}$$

The formulae for the statistical sensitivity measures of the gain, phase and transfer function are [4]:

$$M_\alpha = \sum_{i=1}^{k} \sum_{j=1}^{k} \underset{\sim}{C}_i^t \, \underset{\sim}{K}_1 \, \underset{\sim}{C}_j \, \rho_{ij} \tag{11}$$

$$M_\beta = \sum_{i=1}^{k} \sum_{j=1}^{k} \underset{\sim}{C}_i^t \, \underset{\sim}{K}_2 \, \underset{\sim}{C}_j \, \rho_{ij} \tag{12}$$

$$M_T = M_\alpha + M_\beta \tag{13}$$

and

$$\underset{\sim}{K}_1 = \int_{\omega_1}^{\omega_2} \underset{\sim}{E}_1 \, d\omega \tag{14}$$

$$\underset{\sim}{K}_2 = \int_{\omega_1}^{\omega_2} \underset{\sim}{E}_1 \, d\omega \tag{15}$$

ρ_{ij} is the ith row and jth column component of the covariance matrix $\underset{\sim}{P}$. For the polynomial (10), the matrices $\underset{\sim}{E}_1$ and $\underset{\sim}{E}_2$ are the following:

$$\underset{\sim}{E}_1 = \frac{\omega_o^2 Q^2}{Z} \begin{bmatrix} 4\omega_o^4 Q^2 (\omega-\omega_o)^2 & 2\omega_o^4 (\omega-\omega_o)\omega Q \\ 2\omega_o^4 (\omega-\omega_o)\omega Q & \omega^2 \end{bmatrix} \tag{16}$$

$$\underset{\sim}{E}_2 = \frac{\omega_o^2 Q^2}{Z} \begin{bmatrix} \omega^4 \omega_o^2 & 2\omega^4 (\omega_o-\omega) Q \\ 2\omega^4 (\omega_o-\omega) Q & 4\omega^2 (\omega_o-\omega)^2 Q^2 \end{bmatrix} \tag{17}$$

where

$$Z = [4Q^2 (\omega-\omega_o)^2 + \omega_o^2]^2$$

and

$$\underset{\sim}{C}_i^t = [X_i \frac{\partial U}{\partial X_i} , X_i \frac{\partial W}{\partial X_i}] \tag{18}$$

The deriviations of the matrices $\underset{\sim}{E}_1$, $\underset{\sim}{E}_2$ and $\underset{\sim}{C}_i^t$ are given in [5].

IV. COMPUTER PROGRAM

A program was written whose flowchart of the algorithm is given in Fig. 1. The program enables us to calculate the sensitivity for a particular circuit either with generation or without generation mode [5]. In the generation mode all possible realizations of admittances are considered. The evaluation of the derivatives has been reduced to determining if the element is in the considered product and to assignment of the product value.

As it follows from Fig. 1, the sensitivity can be calculated for different values of circuit parameters which are automatically changing during execution of the program.

V. AN EXAMPLE

In Fig. 2 sensitivity calculations for Tow-Thomas circuit [6] are included. In the calculations the following parameters of op. amps. (3) were assumed:

$BO_i \rightarrow 0.0001$

$BG_i \rightarrow 0.0001, 0.001, 0.01, 0.1, 1, 10$

Note that $\frac{1}{BG_i}$ stands for normalized value of a gain-bandwidth product.

If we take, for example, op. amps. with gain-bandwidth product of 10 MHz, which is typical for good op. amps. commercially available, then the sensitivity measure is calculated for the following values of ω_o: $10^3, 10^4, 10^5, 10^6, 10^7, 10^8$ $[\frac{rd}{sec}]$. For the calculation of sensitivity the following element variation statistics have been assumed:

passive R, C elements

Variance $\rho_{xi}^2 = 10^{-4}$, correlation coefficient $\rho_{ij} = 0.8$

no correlation between R and C

active elements (op. amps.)

variance $\rho_{xi}^2 = 0.4$, correlation coefficient $\rho_{ij} = 0.4$

no correlation between passive and active elements.

The integrals matrices $\underset{\sim}{K}_1$ and $\underset{\sim}{K}_2$ (14), (15) have been calculated over the frequency bands from $\omega = \omega_o(1 - \frac{1}{2R_Q})$ to $\omega = \omega_o(1 + \frac{1}{2R_Q})$. For the circuit under consideration the sensitivity has been calculated for two cases. In the first case, correlation among op. amps. was assumed to be zero (Fig. 2a); while in the second case, non-zero correlation was assumed (Fig. 2b). Having compared the plots we see that the circuit has lower sensitivity for the case without correlation than for the case with correlation.

VI. CONCLUSION

Modification of procedure for Acar-Ghausi statistical sensitivity measure has been presented which includes the effects of non-idealness of the op. amps. The measure takes into account tolerances and correlation of the op. amps. and can be readily calculated for different realizations of circuit admittances. Thus, it can be used in a generation of new low sensitivity active filters. The program has been written to implement this procedure. The sensitivity can be calculated for different values of circuit parameters which can be automatically changed during execution of the program.

ACKNOWLEDGMENT

The author would like to express his gratitude to Dr. M. S. Ghausi for his counsel and advice. The research reported in this paper was supported by the National Science Foundation under Grant ENG-78-07247.

REFERENCES

1. Rosenblum, A., M. Ghausi, Multiparameter Sensitivity in Active RC Networks, IEEE Trans. on Circuits and Systems, Vol. CT-18, November 1971, pp. 592-599.

2. Rosenblum, A., M. Ghausi, Sensitivity Minimization in Active RC Networks, J. of the Franklin Institute, 1972, pp. 95-111.

3. Acar, C., M. Ghausi, Statistical Multiparameter Sensitivity Measure of Gain and Phase Functions, J. Circuit Theory and Application, Vol. 5, 1977, pp. 13-22.

4. Acar, C., M. Ghausi, Statistical Multiparameter Sensitivity Measure in High Q. Networks, J. of the Franklin Institute, 1975, pp. 281-297.

5. Mulawka, J., Computer Generation of Low Sensitivity Biquadratic Active RC Sections via Selection of Admittances, Technical Report, 1979, Oakland University, Rochester, Michigan.

6. Thomas, L., "The Biquad: Part I - Some Practical Design Considerations, Part II - A Multipurpose Active Filtering System," IEEE Trans. Circuit Theory, May 1971, pp. 350-361.

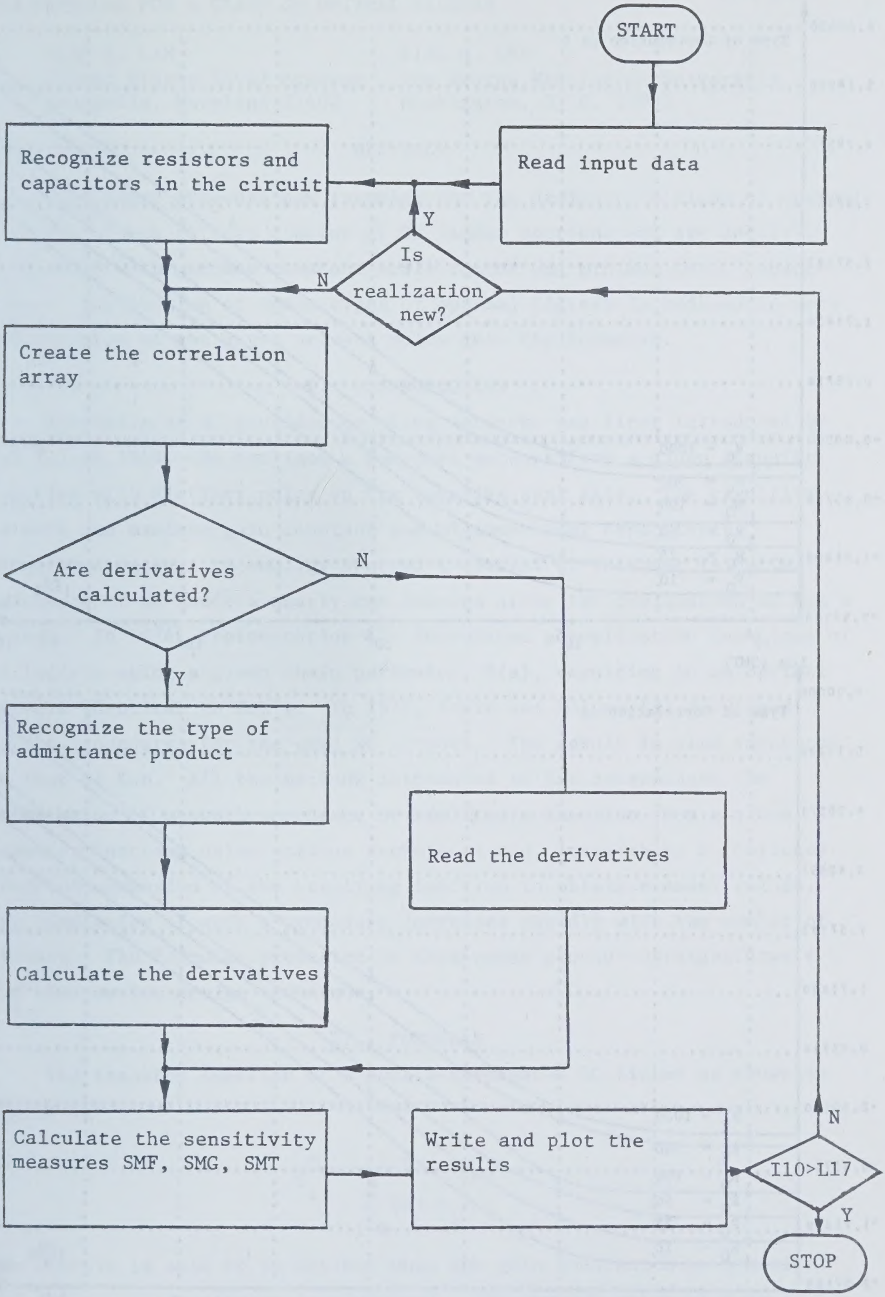

START
Read input data
Recognize resistors and capacitors in the circuit
Is realization new?
Y
N
Create the correlation array
Are derivatives calculated?
N
Y
Recognize the type of admittance product
Read the derivatives
Calculate the derivatives
Calculate the sensitivity measures SMF, SMG, SMT
Write and plot the results
I10>L17
N
Y
STOP

Fig. 1. Flowchart of algorithm

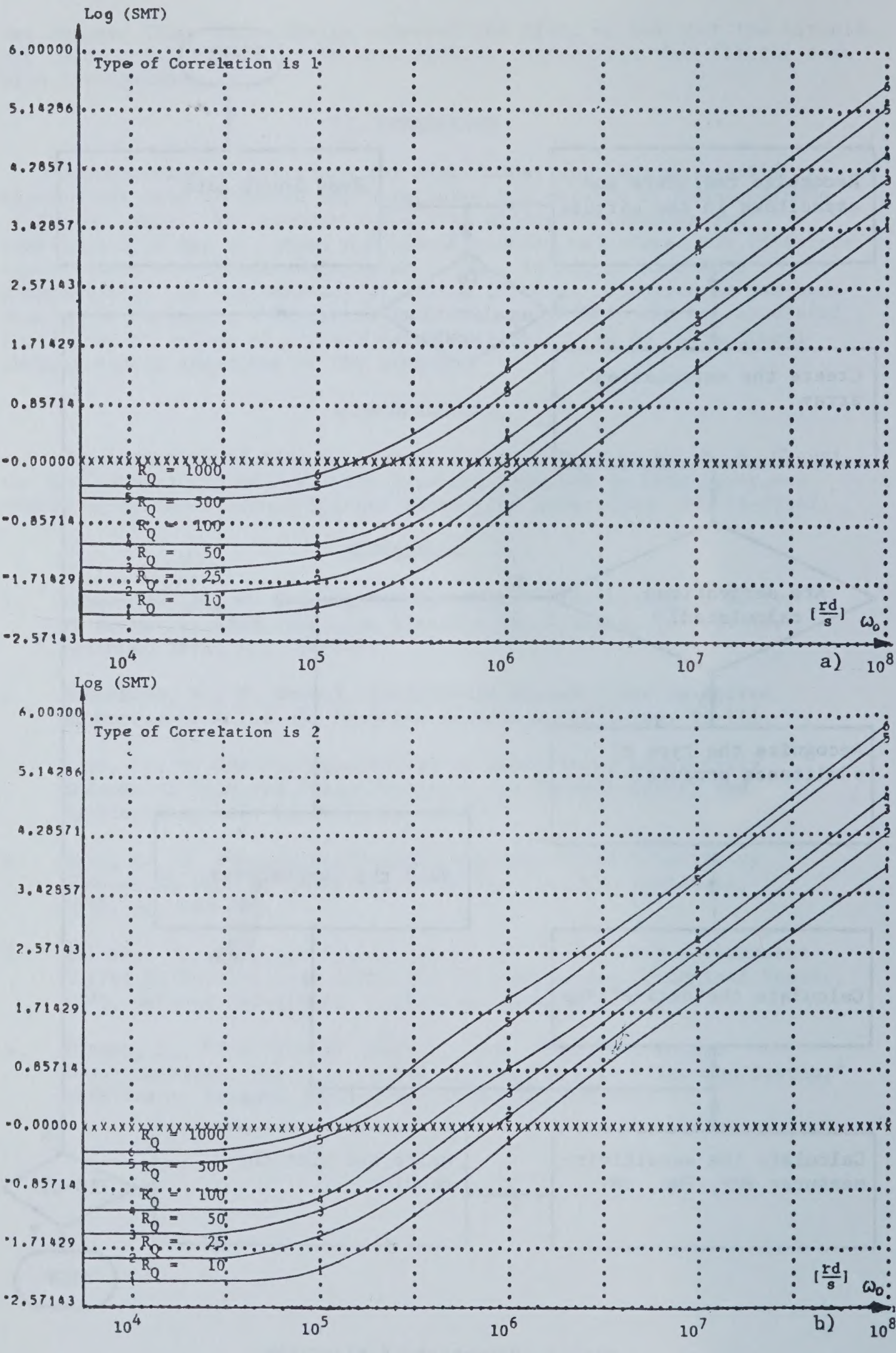


Fig. 2. Sensitivity measure versus center frequency

NEW FORMULAS FOR A CLASS OF OPTIMAL FILTERS

TIAN S. LIM
United States Naval Academy
Annapolis, Maryland 21402

TING N. LEE
The George Washington University
Washington, D. C. 20052

ABSTRACT

This paper presents new formulas for the design of a class of optimal filters. These filters consist of RC ladder sections and are doubly-terminated with maximum constant scale factor and minimum total capacitance. Realization of these types of optimal filters is reduced to mere substitution of the given network poles into the formulas.

I. INTRODUCTION

Synthesis of RC grounded coupling networks was first introduced by Kuh [1] in 1958. He realized a two-port network from a given transfer function with distinct poles on the negative real axis. The resulting network has maximum gain constant and minimum total capacitance. Integrated circuit technology spurred a revival of interest in optimal synthesis of RC ladders nearly two decades after the publication of Kuh's method. In 1974, Protonotarios [2] formulated a realization technique of RC ladders using a given chain parameter, B(s), resulting in an optimal network identical to Kuh's. In 1975, Stein and Salama [3] developed design procedures for the same RC network. The result is also identical to that of Kuh. All the methods introduced so far necessitate the formulation of network impedance or admittance function from a given transfer function using various techniques and, invariably, a continued fraction expansion of the resulting function to obtain element values. The complexity of such a procedure increases rapidly with the number of ladders. The formulas presented in this paper provide straightforward and time-saving design technique.

II. FORMULAS

The transfer function of a double-terminated RC filter as shown in Fig. 1 is

$$\frac{V_o}{V_i} = \frac{H}{\prod_{i=1}^{n} (s+p_i)} \qquad (1)$$

The circuit is said to be optimal when the gain constant H is maximized and the total capacitance is minimized. The maximum value of H is

$$H_{max} = \frac{1}{\prod_{i=1}^{n} C_i \prod_{i=1}^{n+1} R_i} = \frac{R_{n+1}}{\sum_{i=1}^{n+1} R_i} \prod_{i=1}^{n} p_i \, . \tag{2}$$

The existing synthesis methods [1-3] all require tedious process of formulating impedance or admittance function and expanding a continued fraction. This paper presents new formulas as follows.

For a two-section ladder, n=2, it is possible to derive simple and useful formulas:

$$R_2 = \frac{2p_1}{p_2 - p_1} \tag{3}$$

$$C_1 = C_2 = \frac{1}{p_1} \tag{4}$$

For the n=3 case, the short circuit input impedance of the optimal ladder is

$$\frac{1}{y_{11}} = \frac{s^3 + a_2 s^2 + a_1 s + a_0}{s^3 + b_2 s^2 + b_1 s + b_0} \tag{5}$$

The formulas for the circuit element values can be shown, after some lengthy manipulations, to be

$$R_2 = R_3 = \frac{(a_2 - b_2)^2}{b_2(a_2 - b_2) - (a_1 - b_1)} \tag{6}$$

$$C_1 = C_3 = \frac{1}{a_2 - b_2} \tag{7}$$

$$C_2 = \frac{C_1[b_2(a_2 - b_2) - (a_1 - b_1)]^2}{(a_1 - b_1)[b_2(a_2 - b_2) - (a_1 - b_1)] - (a_2 - b_2)[b_1(a_2 - b_2) - (a_0 - b_0)]} \tag{8}$$

where

$a_0 = p_1 p_2 p_3$,

$a_1 = p_1 p_2 + p_1 p_3 + p_2 p_3$

$a_2 = p_1 + p_2 + p_3$,

$$b_0 = \frac{p_1 p_2 p_3 p_{21} p_{31} p_{32}}{p_1 p_2 p_{21} + p_1 p_3 p_{31} + p_2 p_3 p_{32} + p_{21} p_{31} p_{32}} \tag{9}$$

$$b_1 = \frac{p_1^2 p_2^2 p_{21} + p_1^2 p_3^2 p_{31} + p_2^2 p_3^2 p_{32} + p_{21} p_{31} p_{32}(p_1 p_2 + p_1 p_3 + p_2 p_3)}{p_1 p_2 p_{21} + p_1 p_3 p_{31} + p_2 p_3 p_{32} + p_{21} p_{31} p_{32}}$$

$$b_2 = \frac{p_1p_2p_{21}(p_1+p_2)+p_1p_3p_{31}(p_1+p_3)+p_2p_3p_{32}(p_2+p_3)}{p_1p_2p_{21}+p_1p_3p_{31}+p_2p_3p_{32}+p_{21}p_{31}p_{32}}$$

$$+\frac{p_{21}p_{31}p_{32}(p_1+p_2+p_3)}{p_1p_2p_{21}+p_1p_3p_{31}+p_2p_3p_{32}+p_{21}p_{31}p_{32}}$$

$p_{21} = p_2 - p_1$,

$p_{31} = p_3 - p_1$, and

$p_{32} = p_3 - p_2$.

The design procedure using formulas (6), (7), and (8) is demonstrated in Example 2.

For $n > 3$, design formulas have been obtained for the network functions whose poles follow a certain pattern.

In (1), if $p_i = ip_1$, then the synthesis may be effected using the following formulas.

$$R_k = \frac{n(n-1)\cdots(n-k+2)R_1}{(k-1)!} \qquad (9)$$

where $k = 2,3,\cdots,n+1$, and

$$C_k = \frac{2(k-1)!}{n(n-1)\cdots(n-k+1)p_1R_1} \qquad (10)$$

where $k = 1,2,\cdots,n$. Example 3 illustrates the procedure.

III. EXAMPLES

Example 1. Design a filter network with maximum gain constant H given that

$$\frac{V_o}{V_i} = \frac{H}{(s+4)(s+20)} .$$

Formulas (3), (4), and (2) give the following values:

$$R_2 = \frac{2p_1}{p_2 - p_1} = \frac{1}{2}\Omega,$$

$$C_1 = C_2 = \frac{1}{p_1} = \frac{1}{4}F$$

$$H = \frac{R_3p_1p_2}{R_1 + R_2 + R_3} = 32$$

The network is shown in Fig. 2.

Example 2. Design a filter network with maximum gain constant H given the transfer function

$$\frac{V_o}{V_i} = \frac{H}{(s+1)(s+4)(s+8)}$$

Solution: Formulas (2), (6), (7), and (8) yield the following values:

$a_0 = 32$ $a_1 = 44$ $a_2 = 13$

$b_0 = 9.6$ $b_1 = 29.6$ $b_2 = 11.4$

$R_2 = R_3 = 0.667$

$C_1 = C_3 = 0.625F$

$C_2 = 6F$

$H = 0.959$

The completed circuit is shown in Fig. 3.

Example 3. Design an optimal filter given that

$$\frac{V_o}{V_i} = \frac{H}{(s+2)(s+4)(s+6)(s+8)(s+10)} .$$

The terminating resistors are $R_1 = R_6 = 1\Omega$. For n = 5, Equations (9) and (10) yield the following element values:

$$R_2 = R_5 = 5R_1 = 5\Omega$$

$$R_3 = R_4 = 10R_1 = 10\Omega,$$

$$C_1 = C_5 = \frac{2}{5p_1R_1} = \frac{1}{5}F,$$

$$C_2 = C_4 = \frac{1}{10p_1R_1} = \frac{1}{20}\text{ F, and}$$

$$C_3 = \frac{1}{15p_1R_1} = \frac{1}{30}F .$$

The maximum gain is given by Equation (2).

$$H = \frac{R_6}{\sum_{i=1}^{6} R_i} \prod_{i=1}^{5} p_i = 120$$

The network is shown in Fig. 4.

REFERENCES

[1] E. S. Kuh, "Synthesis of RC Grounded Two Ports," IRE Transactions on Circuit Theory, Vol. CT-5, No. 1, pp. 55-61, March 1958.

[2] E. N. Protonotarios, "Optimal Transfer-Function Synthesis of RC Ladders-Lumped and Distributed," IEEE Transactions on Circuits and

Systems, Vol. CAS-21, pp. 49-56, January 1974.

[3] R. A. Stein and A. I. A. Salama, "Resistance and Capacitance Minimization in Low-Pass RC Ladder Networks," IEEE Transactions on Circuits and Systems, pp. 27-31, January 1975.

[4] T. S. Lim, "Synthesis of Optimal Ladder Networks," Doctoral Dissertation, Department of Electrical Engineering and Computer Science, The George Washington University, Washington, D. C., pp. 49-95, May 1977.

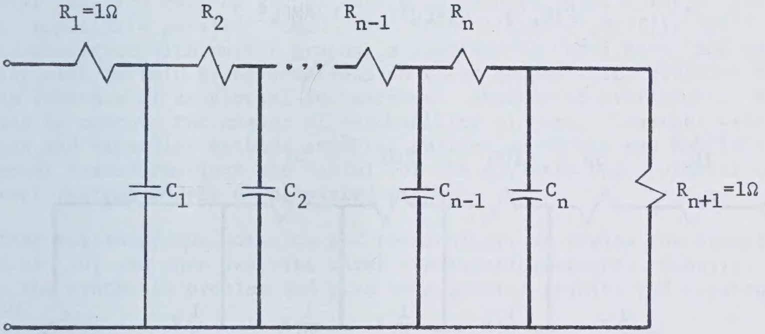

$R_1=1\Omega$
R_2
R_{n-1}
R_n
C_1
C_2
C_{n-1}
C_n
$R_{n+1}=1\Omega$

Fig. 1. Double-Terminated RC Ladder

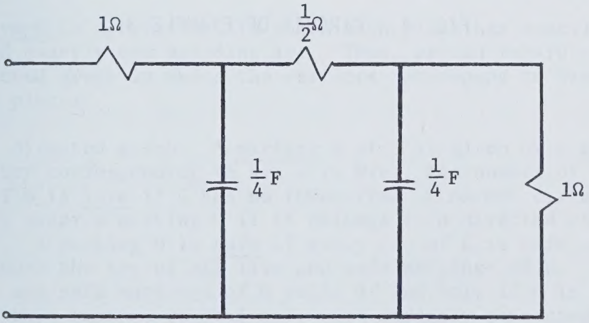

1Ω
$\frac{1}{2}\Omega$
$\frac{1}{4}$F
$\frac{1}{4}$F
1Ω

Fig. 2. Circuit of Example 1

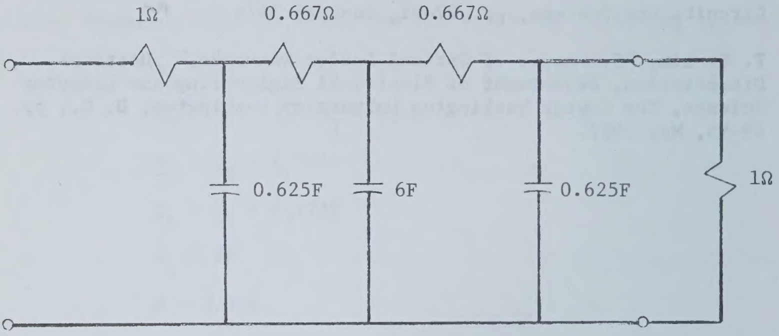

1Ω
0.667Ω
0.667Ω
0.625F
6F
0.625F
1Ω

FIG. 3. CIRCUIT OF EXAMPLE 2

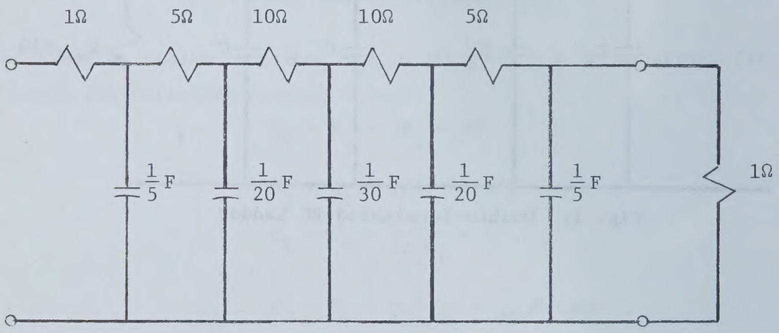

1Ω
5Ω
10Ω
10Ω
5Ω
1/5 F
1/20 F
1/30 F
1/20 F
1/5 F
1Ω

FIG. 4. CIRCUIT OF EXAMPLE 3

ADDITIONAL TRANSFORMATIONS OF LIVE AND SAFE MARKED GRAPHS

R. JOHNSONBAUGH
Dept. of Math.
Chicago State Univ.
Chicago, IL 60628

M.C. KAO and T. MURATA
Dept. of Information Eng.
Univ. of Ill. at Chicago Circle
Chicago, IL 60680

ABSTRACT

This paper is concerned with certain transformations of marked graphs (a subclass of Petri nets) which preserve liveness, safeness, and other properties. Certain synthesis problems are also considered. Together with the reduction and expansion methods reported earlier by Murata and Koh, the present transformations are useful for the analysis and synthesis of parallel computation models using marked graphs.

1. INTRODUCTION

Petri nets and related graphs have been used to model a large class of systems, especially parallel computation methods (see [1,9,11]). This paper is concerned with marked graphs, a subclass of Petri nets, and specifically with certain transformations on these graphs which preserve the liveness (absence of deadlocks) and safeness (absence of overflows). We also wish to compute the number of reachability classes. Together with the reduction and expansion methods reported earlier by Murata and Koh [10], the present transformations are useful for the analysis and synthesis of concurrent systems models using marked graphs.

After outlining our notation and terminology, we review the transformations of [10] and then describe three new transformations. Finally, we discuss the synthesis problem and give some general results and constructions.

2. BASIC TERMINOLOGY

We briefly summarize some definitions and notation concerning marked graphs which we will use throughout this paper. For more complete information on marked graphs the reader is referred to [2,3,5,7,8].

A marked graph is a Petri net in which each place has exactly one incoming arc and exactly one outgoing arc. Thus, we can regard a marked graph as a directed graph in which the vertices correspond to transitions and the arcs to places.

Let G be a directed graph. A marking M of G is given by a column vector whose entry corresponding to arc e is M(e), the number of tokens on e. A marking of G is live if G has no token-free directed circuits. An arc of G is safe under a marking M if it belongs to a directed circuit with token count one. A marking M is safe if every arc of G is safe under M. We let LS(G) denote the set of all live and safe markings of G. It can be shown that live and safe markings of G exist if and only if G is strongly connected. For this reason, we deal only with strongly connected

---

R. Johnsonbaugh was supported by an NSF Faculty Science Professional Development Grant.

T. Murata was supported by NSF Grant ENG 78-05933.

(directed) graphs in this paper. In fact, unless otherwise specified, we will substitute the term "graph" for "strongly connected, directed graph." We will also abbreviate "directed path" to "path."

If G is a graph, we let B_G denote the circuit matrix of all directed circuits in G. Two live markings M and N of G are said to be *equivalent*, denoted $M \simeq N$, if $B_G M = B_G N$. Then $\simeq$ is an equivalence relation. It can be shown that $M \simeq N$ if and only if M and N are mutually reachable (see [3] and [7]). We let $\rho(G)$ denote the number of equivalence classes of live and safe markings of G.

3. TRANSFORMATIONS OF MARKED GRAPHS

In this section we will be concerned with certain transformations of marked graphs. The first type of transformation expands a graph G to a graph G' such that $\rho(G) = \rho(G')$. Moreover, for each such transformation, there is associated a mapping $\psi\colon LS(G') \to LS(G)$, such that if $M' \simeq N'$ in G', then $\psi(M') \simeq \psi(N')$ in G. The inverse relation ψ^{-1} gives a method of constructing live and safe markings in G' given a live and safe marking in G. We summarize three such transformations (Theorems 3.1, 3.2, 3.3) reported by Murata and Koh [10] (see Figs. 1, 2, and 3).

Theorem 3.1. (Parallel transformation) Let G be a graph with arc $e_i = (a,b)$. Let G' be the graph obtained from G by adding an additional (distinct) arc $e_j = (a,b)$. Then $\rho(G) = \rho(G')$.

The parallel transformation and the mapping ψ are shown in Fig. 1. We note that ψ is one-to-one and onto.

Theorem 3.2. (Series transformation) Let G be a graph with arc $e_i = (a,b)$. Let G' be the graph obtained from G by adding an additional vertex c and replacing arc e_i by arcs (a,c) and (c,b). Then $\rho(G) = \rho(G')$.

The series transformation and the mapping ψ are shown in Fig. 2. We note that ψ is onto.

Theorem 3.3. (Unique circuit transformation) Let G be a graph with vertices a and b for which there is a unique path from a to b. Let G' be the graph obtained from G by adding the arc (b,a). Then $\rho(G) = \rho(G')$.

The unique circuit transformation is illustrated in Fig. 3.

The proof of Theorem 7 in [10] shows that the mapping ψ for a unique circuit transformation is obtained by simply deleting the arc (b,a) and its token, if any. We can establish the following properties of ψ.

Theorem 3.4. The mapping ψ for a unique circuit expansion is one-to-one, but is not, in general, onto.

Proof. We use the notation of Theorem 3.3. Let $e = (b,a)$ and let P denote the unique path from a to b. Suppose $\psi(M_1') = \psi(M_2')$ for M_1', $M_2' \in LS(G')$. To prove that ψ is one-to-one we need to show that $M_1'(e) = M_2'(e)$. Since M_1' and M_2' are safe markings, e belongs to a directed circuit with token count one in each of M_1' and M_2'. This circuit must be $P \cup \{e\}$.

Now from the definition of ψ we must have $M_1'(e') = M_2'(e')$ for every $e' \in P$. It follows that $M_1'(e) = M_2'(e)$.

In Fig. 4, $\rho(G) = 1$. There are three directed circuits in G, each of length two, and in each we may mark one arc or the other. This yields eight live and safe markings. There is a unique path from a to b and six markings in which this path has token count zero or one. Hence, G' has six live and safe markings. Therefore, ψ is not onto in this case. $\square$

We call our first additional transformation the Y-V transformation because of the form assumed by the arcs involved. (See Fig. 5.)

Theorem 3.5. (Y-V transformation) Let G be a graph with arcs $e_1 = (a,x)$ and $e_2 = (a,y)$. Let G' be the graph obtained from G by adding an additional vertex b and replacing arcs e_1 and e_2 by arcs $e_1' = (b,x)$, $e_2' = (b,y)$, and $e = (a,b)$. Then $\rho(G) = \rho(G')$. (The conclusion is also valid if we everywhere replace (α,β) by (β,α).)

Proof. The circuit matrices for G and G' can be expressed in the partitioned forms:

$$B_G = \begin{array}{c} \begin{array}{ccc} & e_1 & e_2 \end{array} \\ \left[\begin{array}{ccc} B_0 & 0 & 0 \\ B_1 & 1 & 0 \\ B_2 & 0 & 1 \end{array}\right] \end{array}, \qquad B_{G'} = \begin{array}{c} \begin{array}{cccc} & e_1' & e_2' & e \end{array} \\ \left[\begin{array}{cccc} B_0 & 0 & 0 & 0 \\ B_1 & 1 & 0 & 1 \\ B_2 & 0 & 1 & 1 \end{array}\right] \end{array}.$$

Let $M' \in LS(G')$, say $M' = (M_0 \ \overset{e_1'}{m_1} \ \overset{e_2'}{m_2} \ \overset{e}{m})^T$. We define a marking M of G by

$$M = \begin{cases} (M_0 \quad m_1 \quad m_2)^T & \text{if } m = 0 \\ (M_0 \quad 1 \quad 1)^T & \text{if } m = 1. \end{cases}$$

Then $M \in LS(G)$. Furthermore, if $M', N' \in LS(G')$, then $B_{G'}M' = B_{G'}N'$ if and only if $B_G M = B_G N$. It follows that $\rho(G) = \rho(G')$. $\square$

The mapping ψ for a Y-V transformation is defined in the proof of Theorem 3.5 and is illustrated in Fig. 5. If we reverse the direction of each arrow in Fig. 5 we also obtain a correct diagram. We note that ψ is onto.

The graph G of Fig. 6 contains no series or parallel edges and no unique circuits; thus, no simplification is possible using the transformations of [10] (Theorems 3.1, 3.2, and 3.3). However, we may apply the Y-V transformation as indicated in Fig. 6 to transform G to G' which is the union of two graphs G_1' and G_2'. Our next transformation will allow us to immediately compute $\rho(G) = \rho(G') = 4$.

The next transformation differs from those discussed in that $\rho(G)$ may not remain constant. However, $\rho(G)$ is computable in terms of subgraphs.

Of course, if $\rho(G_1) = 1$ or $\rho(G_2) = 1$, $\rho(G)$ remains constant in this case also.

Theorem 3.6. (Separable graph transformation) Let G_1 and G_2 be graphs with exactly one vertex in common. If $G = G_1 \cup G_2$, then $\rho(G) = \rho(G_1)\rho(G_2)$.

Proof. The circuit matrix for G may be partitioned

$$B_G = \begin{bmatrix} B_{G_1} & 0 \\ 0 & B_{G_2} \end{bmatrix}.$$

If $M_i \varepsilon LS(G_i)$, $i = 1,2$, then $M = (M_1\ M_2)^T \varepsilon LS(G)$. Moreover, $B_{G_i} M_i = B_{G_i} N_i$, $i = 1,2$, if and only if $B_G M = B_G N$. The conclusion follows. []

In Fig. 6, G' is separable, hence $\rho(G') = \rho(G_1')\rho(G_2')$. Of course, G_1' and G_2' are isomorphic, so $\rho(G') = \rho(G_1')^2$. The graph G_1' is known as the necklace on three vertices (see Fig. 7). In general the necklace on k vertices N_k is defined as the graph with k vertices $v_0, v_1, \ldots, v_{k-1}$ and 2k arcs which exist from v_i to v_j whenever $i-k = \pm 1 \pmod{k}$. In [2] it is shown that $\rho(N_k) = k-1$. In Fig. 7 representatives of the two equivalence classes of $LS(N_3)$ are illustrated. It now follows that in Fig. 6 $\rho(G) = \rho(G') = \rho(G_1')^2 = 4$.

The mapping ψ for a separable graph transformation takes a different form from the mappings associated with the previous transformations. For a separable graph transformation ψ is defined as the mapping from LS(G) into $LS(G_1) \times LS(G_2)$ whose inverse is given by $\psi^{-1}(M_1,M_2) = (M_1\ M_2)^T$. In this case ψ is one-to-one and onto. As an example we construct a member of the four equivalence classes of live and safe markings for the graph G of Fig. 6. Two markings are illustrated in Fig. 8. We begin with markings M_1 and M_2 of Fig. 7, then form $(M_1\ M_1)^T$, $(M_1\ M_2)^T$, $(M_2\ M_1)^T$, and $(M_2\ M_2)^T$ for G' of Fig. 6. We then use ψ^{-1} for the Y-V transformation to obtain the four markings for G.

Our final transformation is similar to the separable graph transformation in that it combines two graphs.

Theorem 3.7. (Transformation by unique paths) Let G_1 and G_2 be graphs with only vertices a and b in common and with no arcs in common. Suppose that there exists a unique path P_i in G_i from a to b, $i = 1,2$. Let $G = G_1 \cup G_2 \cup \{e\}$ where $e = (b,a)$. For each equivalence class $\overline{M}_{i,j} \subseteq LS(G_i)$, $j = 1, \ldots, \rho(G_i)$, let $m_{i,j}$ denote the number of distinct non-zero token counts possible on P_i, $i = 1,2$. Then

$$\rho(G) = \rho(G_2) \sum_{j=1}^{\rho(G_1)} m_{1,j} + \rho(G_1) \sum_{j=1}^{\rho(G_2)} m_{2,j} - \rho(G_1)\rho(G_2).$$

Proof. Fix an element $\overline{M}_2 \varepsilon LS(G_2)$ and an integer j, $1 \leq j \leq \rho(G_1)$. We construct $m_{1,j}$ distinct equivalence classes of LS(G).

It is possible to successively fire the vertices of P_2 to obtain a marking $M_2 \varepsilon \overline{M}_2$ under which P_2 has token count one. Consider an element $M_{1,j} \varepsilon \overline{M}_{1,j}$, where the path count of P_1 under $M_{1,j}$ is non-zero. Define a marking M of G as $M(e) = 0$, $M(e') = M_i(e')$ for $e' \neq e$, $i = 1,2$.

We first show that M is live. The arc e belongs only to the two directed circuits, $P_1 \cup \{e\}$ and $P_2 \cup \{e\}$, each of which has positive token count. Obviously, any directed circuit wholly contained in either G_1 or G_2 has a positive token count since the markings M_1 and M_2 are live. Thus, consider a directed circuit $C \subseteq G_1 \cup G_2$ which has arcs in both G_1 and G_2. Either $C \cap G_1 = P_1$ or $C \cap G_2 = P_2$; hence, C has a positive token count. Therefore, M is live.

The arc e is safe since the directed circuit $P_2 \cup \{e\}$ has token count one. All other arcs are safe since M_1 and M_2 are safe. Therefore, $M \varepsilon LS(G)$.

If we choose another element $M'_{1,j} \varepsilon \overline{M}_{1,j}$, with a different token count on P_1, we will obtain a member of a different equivalence class of LS(G). Thus, we obtain $m_{1,j}$ equivalence classes of LS(G). If we choose another element $M^*_2 \varepsilon \overline{M}^*_2 \neq \overline{M}_2$ with P_2 having token count one, we obtain $m_{1,j}$ additional equivalence classes of LS(G). We have constructed a total of $\rho(G_2)m_{1,j}$ equivalence classes of LS(G). Summing over all equivalence classes of G_1 we obtain

$$\rho(G_2) \sum_{j=1}^{\rho(G_1)} m_{1,j}$$

equivalence classes of LS(G).

The roles of G_1 and G_2 may be reversed to obtain

$$\rho(G_1) \sum_{j=1}^{\rho(G_2)} m_{2,j}$$

equivalence classes of LS(G).

Since we have counted the equivalence classes of LS(G) with paths P_1 and P_2 each having token count one twice, we actually have counted

$$\rho(G_2) \sum_{j=1}^{\rho(G_1)} m_{1,j} + \rho(G_1) \sum_{j=1}^{\rho(G_2)} m_{2,j} - \rho(G_1)\rho(G_2)$$

distinct equivalence classes of LS(G).

To conclude the proof we must show that we have counted all the equivalence classes of LS(G). To this end, let $M \in \overline{M}$, an equivalence class of LS(G). We may assume that arc e is token-free. Since the token-free arc e belongs only to the two directed circuits, $P_1 \cup \{e\}$ and $P_2 \cup \{e\}$, and M is safe, the token count of either P_1 or P_2 is one.

Define a marking M_i on G_i, i = 1,2, by restriction. Clearly, M_i is live, i = 1,2. We show that M_i is also safe, i = 1,2, which will complete the proof, since $\overline{M}$ would have been counted as having been generated by M_1 and M_2. The argument is adapted from the proof of Theorem 7 of [10].

Let i = 1 or i = 2. The only arc that might be unsafe under M_i is one in P_i. Suppose $e_j = (c,d) \in P_i$ is unsafe. Since G_i is strongly connected, there exists a directed circuit C containing e_j. Since e_j is unsafe, the token count of C is greater than one. Since M is safe for G every arc in $C-P_i$ is safe. Thus, each arc in $C-P_i$ is contained in a directed circuit C_k with token count one. Since e_j is unsafe under M_i, no C_k can contain e_j. Now the union of $\{C_k - \{e_j\}\}_k$ gives a path from c to d distinct from $e_j = (c,d)$. This contradicts the uniqueness of P_i. Therefore, M_i is safe. □

As an application of Theorem 3.7, we prove that $\rho(N_k) = k-1$. The proof for the case k = 3 is illustrated in Fig. 9.

Theorem 3.8. For the necklace N_k, we have $\rho(N_k) = k-1$.

Proof. Let v_0 and v_1 be consecutive vertices of N_k. Let $G_2 = \{(v_0,v_1),(v_1,v_0)\}$ and $G_1 = N_k - G_2$. Then there is a unique path P_i in G_i from v_0 to v_1, i = 1,2. Now $\rho(G_1) = 1 = \rho(G_2)$, thus we must compute $m_{1,1}$ (respectively $m_{2,1}$), the number of distinct non-zero token counts possible on P_1 (respectively P_2). For G_2, we can only have the token count equal to 1 on P_2, hence $m_{2,1} = 1$. On P_1, the token counts 1, 2, ..., k-1 are possible (see Fig. 9), hence $m_{1,1} = k-1$. By Theorem 3.7,

$$\begin{aligned}\rho(N_k) &= \rho(G_2)m_{1,1} + \rho(G_1)m_{2,1} - \rho(G_1)\rho(G_2) \\ &= 1 \cdot (k-1) + 1 \cdot 1 - 1 \cdot 1 \\ &= k-1. \qquad \square\end{aligned}$$

The mapping ψ from LS(G) into $LS(G_1) \times LS(G_2)$ for the transformation by unique paths is given by restriction. This mapping is one-to-one but, in general, is not onto. The argument is essentially the same as that given for the unique circuit transformation.

The transformation by unique paths permits reductions and expansions not obtainable using the other five transformations. The graph G of Fig. 10 contains no series edges, no parallel edges, no Y-V configuration, and no edge lies in a unique circuit. Also, G is not separable. However, G can be simplified using the transformation by unique paths as indicated in Fig. 10. In each of G_1 and G_2 there is a unique path from d to c. As shown in the diagram, G_2 can be (series) transformed to the necklace N_3. Thus, $\rho(G_2) = 2$. Also, $\rho(G_1) = 1$, $m_{1,1} = 3$, $m_{2,1} = 1$, and $m_{2,2} = 1$. Therefore,

$$\rho(G) = 2 \cdot 3 + 1 \cdot (1+1) - 1 \cdot 2 = 6.$$

4. SYNTHESIS

In this section we will use the transformations of the previous section to investigate certain synthesis problems. Some synthesis problems were also considered in [7] and [9].

We wish to construct a marked graph satisfying certain given conditions such as the number of equivalence classes of live and safe markings, the number of arcs, and the number of vertices. Throughout this section, in a given graph, we let p denote the number of arcs and t denote the number of vertices.

In an example in [10], a graph G was constructed with $\rho(G) = 1$, $p = 16$, and $t = 9$. We give a general construction in case $\rho(G) = 1$.

Theorem 4.1. Given positive integers $i \geq j \geq 2$, there exists a graph G with $\rho(G) = 1$, $p = i$, $t = j$.

Proof. Let G_1 be a directed circuit with j vertices and j arcs (see Fig. 11). Clearly $\rho(G_1) = 1$. If $i = j$, take $G = G_1$. Otherwise, insert $i-j$ parallel arcs (see Fig. 11) to obtain a graph G. By Theorem 3.1, $\rho(G) = 1$. Since $p = i$ and $t = j$, the construction is complete. ▯

Theorem 4.1 is the best possible since if $p < t$, G is either a tree, and thus is not strongly connected, or disconnected. (See [4], page 42.)

For $\rho(G) > 1$, we can prove the following result by methods similar to those used in the proof of Theorem 4.1.

Theorem 4.2. Let i and j be integers satisfying $i-j \geq n+1 \geq 3$ and $j \geq n+1$. Then there exists a graph G satisfying $\rho(G) = n$, $p = i$, $t = j$.

Proof. Let N_{n+1} be the necklace on n+1 vertices. Then $\rho(N_{n+1}) = n$. Add $j-n-1$ edges in series to N_{n+1} to obtain G_1 (see Fig. 12). Now G_1 has $j+n+1$ edges and j vertices. Next, add $i-j-n-1$ edges in parallel to obtain G (see Fig. 12). By Theorems 3.1 and 3.2, $\rho(G) = n$ and G has the required properties. ▯

We note that the proofs of Theorems 4.1 and 4.2 which give the construction of the desired graph G may be combined with the transformations ψ of Section 3 to obtain the markings for G.

Theorem 4.2 is not the best possible result since the restriction $j \geq n+1$ is too severe. For example, for K_4, the complete graph on four vertices, we have $\rho(K_4) = 3! = 6$ (see [2]), yet $j = 4 < 7 = n+1$.

We close this section by noting that in some cases we may apply Theorem 3.6 to obtain a graph G with prescribed $\rho(G)$ with fewer vertices than that given by simply taking G to be an appropriate necklace. For example, suppose we want $\rho(G) = 8$. The necklace N_9 has nine vertices, but using Theorem 3.6 we can easily construct a graph G with seven vertices for which $\rho(G) = 8$. We simply let G be obtained by joining three copies of N_3 as shown in Fig. 13. By Theorem 3.6, $\rho(G) = \rho(N_3)^3 = 8$.

5. CONCLUSIONS

Three additional transformations of marked graphs have been introduced which, together with those introduced by Murata and Koh [10], are useful for analysis and synthesis of marked graphs. The examples following Theorems 3.5 and 3.8 show that these new transformations are extensions of the earlier transformations in that the new ones apply to a larger class of graphs than the old ones did.

These transformations have been shown to be useful for reducing a marked graph to a simpler one while either preserving certain properties or providing a means of computing the changes in the properties. We have also shown how to transform live and safe markings as the graphs are transformed. The reduction methods are useful for computing $\rho(G)$ for a graph G, especially since a general method of obtaining $\rho(G)$ is unknown. On the other hand, we can also use these transformations to construct a marked graph from simpler ones to achieve certain given conditions. In particular, we have dealt with the features: construction of live and safe markings, the number of reachability classes, the number of arcs, and the number of vertices. In [6] Johnsonbaugh dealt with the questions of resource requirements and the total number of live and safe markings. These results will be published elsewhere. The transformations used in our synthesis methods always lead to live and safe markings whereas some previous methods (for example [7]) did not guarantee safe markings.

We have shown how the transformations can be used to reduce or expand graphs. It would be useful to have a complete set of transformations, that is, a set of transformations which could transform any graph G_1 to any other graph G_2 if $\rho(G_1) = \rho(G_2)$. In any case, it would be interesting to find additional transformations under which ρ is invariant.

REFERENCES

[1] T. Agerwala, "Some applications of Petri nets," Procs. of NEC '78, Chicago, IL, Vol. 32, pp. 149-154, Oct. 1978.

[2] F. Commoner, et al., "Final report for the project-development of theoretical foundations for description and analysis of discrete information systems," Vol. II-Mathematics, CADD-7405-2011, Mass. Comp. Assoc., Inc., Wakefield, MA, May 1974.

[3] F. Commoner, A.W. Holt, S. Even, and A. Pnueli, "Marked directed graphs," J. of Comp. and Syst. Sci., Vol. 5, pp. 511-532, Oct. 1971.

[4] N. Deo, Graph Theory with Applications to Engineering and Computer Science, Prentice-Hall, Englewood Cliffs, NJ, 1974.

[5] A.W. Holt and F. Commoner, "Events and conditions," Record of the Project MAC Conf. on Concurrent Systems and Parallel Computation, ACM, N.Y., pp. 3-52, 1970.

[6] R. Johnsonbaugh, "Transformations of live and safe marked graphs," M.S. thesis, Dept. of Inf. Eng., Univ. of Ill., Chicago, IL, 1979.

[7] T. Murata, "Circuit theoretic analysis and synthesis of marked graphs," IEEE Trans. on Circuits and Systems, Vol. CAS-24, No. 7, pp. 400-405, July 1977.

[8] T. Murata, "Petri nets, marked graphs, and circuit-system theory -- a recent CAS application," Circuits and Systems, Vol. 11, No. 3, pp. 2-12, June 1977.

[9] T. Murata, "Synthesis of marked graph computation models for prescribed resources and performance," to appear in Procs. of COMPSAC '79.

[10] T. Murata and J.Y. Koh, "Reduction and expansion of live and safe marked graphs," Procs. of the 1979 KSCAS/Tokyo, also to appear in the IEEE Trans. on Circuits and Systems.

[11] J.L. Peterson, "Petri nets," Computing Surveys, Vol. 9, No. 3, pp. 223-252, Sept. 1977.

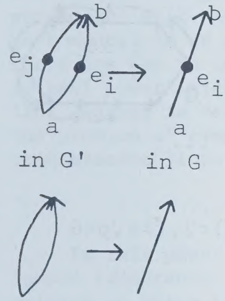


FIG. 1.

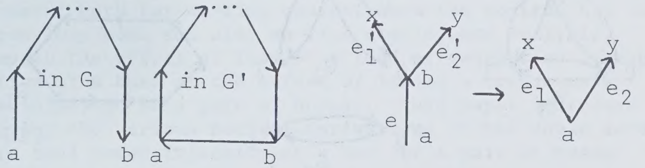


FIG. 3.

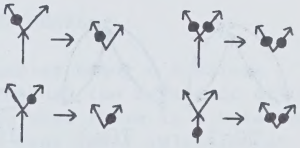


FIG. 5.

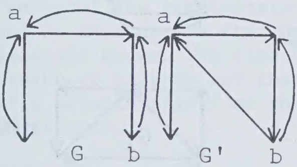


FIG. 4.

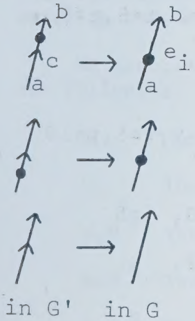
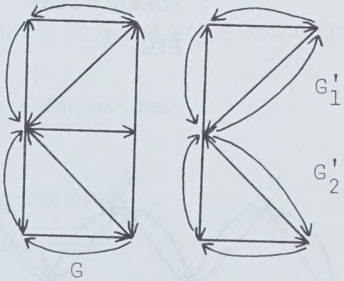


FIG. 2.



$G' = G_1' \cup G_2'$

FIG. 6.

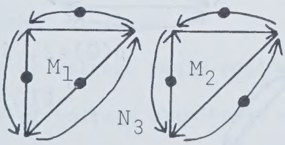


FIG. 7.

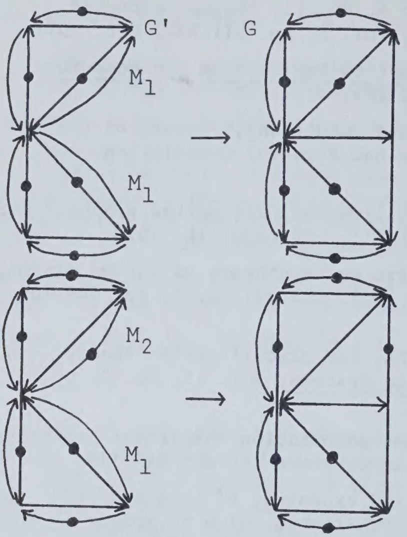

G'
M1
M1
M2
M1
G

FIG. 8.

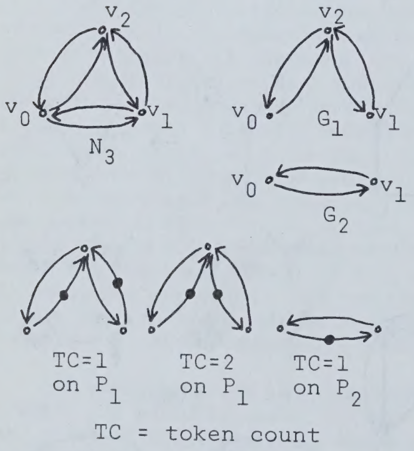

v2
v0
v1
N3
v2
v0
G1
v1
v0
v1
G2
TC=1
on P1
TC=2
on P1
TC=1
on P2
TC = token count

FIG. 9.

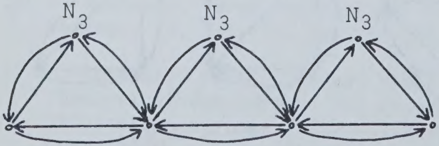


FIG. 13.

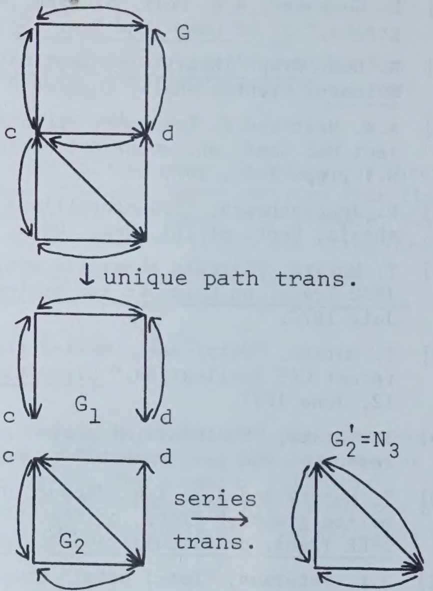

G
c
d
unique path trans.
c
G1
d
c
d
G2
series
trans.
G'2=N3

FIG. 10.

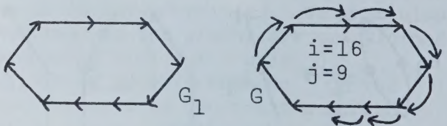
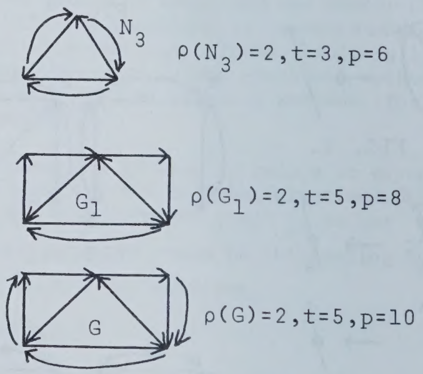


FIG. 11.


N3
ρ(N3)=2,t=3,p=6
G1
ρ(G1)=2,t=5,p=8
G
ρ(G)=2,t=5,p=10

$\rho(G)=2$, $i=10$, $j=5$

FIG. 12.

COMPUTATION OF THE EFFECT OF REAL POWER INJECTION ON THE ANGLE ACROSS TRANSMISSION LINES IN POWER NETWORKS

NATARAJAN NARASIMHAMURTHI
Washington University
Department of Systems Science and Mathematics
Box 1040
St. Louis, Missouri 63130

0. ABSTRACT

In this paper we provide a systematic method for obtaining the n-th partial derivative of the angles across transmission lines with respect to power injection at a bus (or a pair of buses) using the first (n-1) partial derivatives. The attractiveness of this scheme is that (1) the same linear network is solved for each derivative (with different injections), and (2) the injection at each bus can be computed in a decentralized manner. The derivation, which uses elementary calculus, can be generalized to a wide class of nonlinear networks.

1. INTRODUCTION

In the analysis of power systems, one of the variables of primary interest is the angle across transmission lines. These angles are controlled by controlling the real power injections into the transmission network. Thus it is important to have efficient computational procedures to determine the effect of a given power injection on the angles across lines. Such procedures will be useful both for on-line control when the control has to be computed in an open loop mode and also on-line contingency analysis, where one has to compute the effect of loss of a load or generation (change in real power injection at a bus) or the effect of losing a transmission line (change in real injection at a pair of buses). This paper addresses the problem of computing the various partial derivatives of the angle across transmission lines to real power injection at a bus (or a pair of buses)

2. NOTATIONS, ASSUMPTIONS AND THE MODEL

In this paper we model the transmission lines as being a lossless lumped inductance and we also assume that at each bus the magnitude of the voltage as being fixed and known. We also assume that there is a reference bus whose phase is taken to be zero and that the real power injection at this bus automatically adjusts itself so that the sum of the real powers entering the network is zero.

We associate with the transmission network a directed graph and define the following:

N: The number of buses excluding the reference bus

b: The number of lines

$\theta \epsilon \mathbb{R}^N$: Vector of voltage phase angles at the buses

$E \epsilon \mathbb{R}^N$: Vector of voltage magnitudes at the buses

$P \epsilon \mathbb{R}^N$: Vector of Real power injection at the buses

$p \varepsilon \mathbb{R}^b$: Vector of Real power flowing on the lines

$\psi \varepsilon \mathbb{R}^b$: Vector of phase angles across lines

$A \varepsilon \mathbb{R}^{Nxb}$: The reduced incidence matrix

$F \varepsilon \mathbb{R}^b$: Vector defined as

$$F_k = \frac{E_i E_j}{x_k} \tag{1}$$

Where line k goes from bus i to bus j and x_k is the inductance of the line

$g : \mathbb{R}^b \to \mathbb{R}^b$ given by

$$g_k(\psi) = F_k \sin(\psi_k) \tag{2}$$

Simple calculations show that

$$p = g(\psi) \tag{3}$$

ie $\quad p_k = F_k \sin(\psi_k)$

Thus the mathematical model for the transmission network model is given by:

$$A\,p = P \tag{4}$$

$$p = g(\psi) \tag{5}$$

$$\psi = A^T \theta \tag{6}$$

3. STATEMENT OF THE PROBLEM

We shall state the problem in a more general setting then mentioned in the introduction.

Suppose that we have an inititial solution (P_0, θ_0, ψ_0) to equation 4, 5 and 6 and we assume that the map $P \mapsto \theta$ is locally invertible at (P_0, θ_0). Let $\psi(x)$ be the solution to the network with the injection given by

$$P = P_0 + P_1 x$$

around ψ_0. We are required to obtain

$$\frac{d\psi}{dx}, \frac{d^2\psi}{dx^2} \cdots \frac{d^n\psi}{dx^n} \tag{7}$$

evaluated at x = 0.

Remark: A sufficient condition for the assumption to hold is that the Jacobian

$$J = A \frac{dg}{d\psi} (\psi_0) A^T$$

be non singular. It can be shown [1] that this is also a necessary condition.

4. SOLUTION TO THE PROBLEM

4.1 Preliminaries.

By linearization of the network equations we mean that equation (5) is replaced by

$$p = \frac{dg}{d\psi} (\psi_0) \psi \tag{5L}$$

Also as a book keeping device we introduce the following definition:

We say that $\hat{p}$ is a flow for the injection $\underline{P}$ if it satisfies

$$A\hat{p} = \underline{P}$$

It should be emphasized that $\hat{p}$ has no physical significance and is introduced here mainly because in what follows it is easier to specify $\hat{p}$ than P.

Since g is a diagonal function we define the various derivatives of g denoted by g', g", g''' ... $g^{(n)}$ as diagonal matrices defined as

$$g^{(k)}_{i,i} (\psi) = \frac{\partial^k g_i (\psi)}{\partial \psi_i^k}$$

4.2 Computation of $\frac{d\psi}{dx}$.

Differentiating 4, 5 and 6 with respect to x we get

$$A \frac{dp}{dx} = P_1 \tag{8}$$

$$\frac{dp}{dx} = g'(\psi(0)) \frac{d\psi}{dx} \tag{9}$$

$$\frac{d\psi}{dx} = A^T \frac{d\theta}{dx} \tag{10}$$

In other words

$\frac{d\psi}{dx}$ is obtained by solving the linearized equations with injection P_1

4.3 Computation of $\frac{d^2\psi}{dx^2}$.

Combining 8 and 9 we get

$$A\, g'(\psi)\Big|_{\psi_0} \frac{d\psi}{dx}(0) = P_1$$

and differentiating this with respect to x we get

$$A\, g'(\psi_0)\frac{d^2\psi}{dx^2} + A\, g''(\psi)\Big|_{\psi_0} \operatorname{diag}\left(\frac{d\psi}{dx}(0)\right)\frac{d\psi(0)}{dx} = 0 \tag{11}$$

and

$$\frac{d^2\psi}{dx^2} = A^T \frac{d^2\theta}{dx^2} \tag{12}$$

Equation (11) and (12) can be interpreted as

$\frac{d^2\psi}{dx^2}$ is obtained by solving the linearized equations with a flow $\hat{p}^{(2)}$ given by

$$\hat{p}_i^{(2)} = -\, g_i''(\psi(0))\left(\frac{d\psi_i}{dx}\right)^2 \tag{13}$$

4.4 Computation of higher order derivatives.

Differentiating (11) with respect to ψ we can obtain interpretations analogous to (13). For example:

$\frac{d^3\psi}{dx^3}$ is obtained by solving the linearized equations with a flow $\hat{p}^{(3)}$

given by

$$-\,\hat{p}_i^{(3)} = g_i'''(0)\left(\frac{d\psi_i}{dx}\right)^3 + 3\, g_i''(0)\,\frac{d\psi_i}{dx}\,\frac{d^2\psi_i}{dx^2} \tag{14}$$

Remarks: 1) To obtain higher order derivatives, the formulae for the flow become tedious to compute but the form of the interpretations remain the same.

2) To formally obtain (13), (14) and higher order terms we need to differentiate $g(\psi(x))$ repeatedly with respect to x. For example

$$\frac{dg}{dx} = \frac{dg}{d\psi}\,\frac{d\psi}{dx}$$

$$\frac{d^2g}{dx^2} = \left[\frac{d}{dx}\left(\frac{dg}{d\psi}\right)\right]\frac{d\psi}{dx} + \frac{dg}{d\psi}\,\frac{d}{dx}\,\frac{d\psi}{dx}$$

$$= \left[\frac{d^2g}{d\psi^2}\,\frac{d\psi}{dx}\right]\frac{d\psi}{dx} + \frac{dg}{d\psi}\,\frac{d^2\psi}{dx^2}$$

$$\frac{d^3g}{dx^3} = \frac{d}{dx}\left(\frac{d^2g}{d\psi^2}\right)\left(\frac{d\psi}{dx}\right)^2 + \frac{d^2g}{d\psi^2}\cdot 2\,\frac{d\psi}{dx}\,\frac{d^2\psi}{dx^2}$$

$$+ \frac{d}{dx}\left(\frac{dg}{d\psi}\right)\frac{d^2\psi}{d\psi^2} + \frac{dg}{d\psi}\,\frac{d^3\psi}{dx^3}$$

$$= \frac{d^3g}{d\psi^3}\left(\frac{d\psi}{dx}\right)^3 + 3\,\frac{d^2g}{d\psi^2}\,\frac{d^2\psi}{dx^2}\,\frac{d\psi}{dx} + \frac{dg}{d\psi}\,\frac{d^3\psi}{dx^3}$$

etc.

and the negative of the flow can be directly written down from the second and higher order terms.

5. CONCLUDING REMARKS

The results in this paper can be viewed as purely algebraic manipulation of the basic network equations. Its importance rests on the network interpretation of the results. In most approximate analyses of power system only the first derivative is computed. The results in this paper show that computation of additional derivatives require little additional effort, as we have to solve the same linear network. As the size of the netowrk that are analyzed is continually increasing, diakoptic and tearing methods [2] will gain prominance. The results in this paper are ideally suited for such methods, as the flow in each line is defined in terms of the various derivatives of the angle across the same line. Though for the sake of theoretical completion we have modelled all lines by their non-linear characteristics, it may be sufficient to model lines sufficiently far from the busses where the power injection changes as being linear.

Equations (4), (5) and (6) can also be interpreted as the network equations of a general non linear resistive network. Since the derivation depends only on the fact that g is a diagonal function, the results can be applied to any nonlinear, uncoupled resistive networks.

6. ACKNOWLEDGEMENT

This research was supported by the U.S. Department of Energy under Control No. ET 78-D-01-3090.

7. REFERENCES

1. Narasimhamurthi, N. "A Note on the Local Invertibility of Load Flow Equations", Report SSM 7909, Department of Systems Science and Mathematics, Washington University, St. Louis, Missouri 63130.

2. Happ, H. H. "Diakoptics and Piecewise Methods", IEEE Transactions on Power Apparatus and Systems, Vol. PAS-89, No. 7, pp. 1373-1382, September/October 1970.

POWER SYSTEM STABILITY INCLUDING NON-TRIVIAL TRANSFER CONDUCTANCES

S. J. Skar
Department of Mathematics, Iowa State University
Ames, Iowa 50011

ABSTRACT

This paper gives criteria for the existence, uniqueness, and stability of synchronous states of power systems including non-trivial transfer conductances. An example of an unstable synchronous state with $|\delta_i - \delta_j - \alpha_{ij}| < \pi/2$ for all rotor angle pairs δ_i, δ_j and all transfer admittance phases $\alpha_{ij} + \pi/2$ and an example of a stable synchronous state with rotor angles 180^o out of phase are given.

I. INTRODUCTION

Nearly all work in the transient stability of power systems uses a model neglecting transfer conductances [2], [7]. In this model a synchronous state for which the rotor angles are less than 90^o out of phase is known to be stable. It is usually assumed that an analogous property holds if transfer conductances are included. An example in this paper contradicts this assumption, and analytic criteria for the stability and instability of synchronous states with non-trivial transfer conductances are given. In addition, results on the existence and uniqueness of synchronous states and an example of a stable synchronous state with rotor angles 180^o out of phase are given. The results are proved in [5] and [6]. Only a brief summary of the procedure will be given here.

II. THE SWING EQUATIONS

For the study of transient stability a system of n synchronous generators is usually modeled by the swing equations [2], [7]

$$M_i\ddot{\delta}_i + d_i\dot{\delta}_i = P_i - \sum_{\substack{j=1\\ j\neq i}}^{n} P_{ij}\sin(\delta_i - \delta_j - \alpha_{ij}) \tag{1}$$

for i=1, ..., n where δ_i is the i-th rotor angle and the slowly varying quantities $M_i > 0$, $d_i > 0$, P_i, $P_{ij} = P_{ji} > 0$, and $\alpha_{ij} = \alpha_{ji}$ are considered to be constant for i, j=1, ..., n, i≠j. The α_{ij} are the transfer conductance parameters and are usually set equal to zero even though they may not be negligible [2].

For the i-th generator M_i is the inertia constant, d_i is the damping constant, and $P_i = Q_i - E_i^2 Y_{ii}\cos\theta_{ii}$, $P_{ij} = E_iE_jY_{ij}$, $\alpha_{ij} = \theta_{ij} - \pi/2$ for i, j=1, ..., n, i≠j where Q_i is the mechanical power, $E_i \exp(\delta_i \;\; -1)$ is the voltage behind the transient reactance, and $Y_{ij}\exp(\theta_{ij}\sqrt{-1})$ is the transfer admittance between machines i and j.

Let $\delta = (\delta_1, \ldots, \delta_n)$ represent a solution of system (1). A solution δ^* is called a synchronous state of (1) if $\dot{\delta}_i^* = \omega_o$ is constant for i=1, ..., n. A synchronous state can be written in the form

$$\delta_i^* = \omega_o t + \sigma_i^* \tag{2}$$

where σ_i^* is some constant for i=1, ..., n. Note that $\delta_i^* - \delta_j^*$ is constant for i, j=1, ..., n. We say that two synchronous states δ^* and δ^{**} are equivalent if there is a constant δ_o such that $\delta_i^{**} = \delta_i^* + \delta_o$ for i=1, ..., n. That is, two synchronous states differing by a constant phase are considered equivalent.

Using the transformation

$$\omega_i = \dot{\delta}_i, \quad i=1, \dots, n \tag{3a}$$

$$\sigma_i = \delta_i - \delta_n, \quad i=1, \dots, n \tag{3b}$$

$$\rho = \delta_n \tag{3c}$$

we can write system (1) in the equivalent form

$$M_i\dot{\omega}_i = -d_i\omega_i + P_i - \sum_{\substack{j=1 \\ j\neq i}}^{n} P_{ij}\sin(\sigma_i - \sigma_j - \alpha_{ij}), \quad i=1, \dots, n \tag{4a}$$

$$\dot{\sigma}_i = \omega_i - \omega_n, \quad i=1, \dots, n-1 \tag{4b}$$

$$\dot{\rho} = \omega_n \tag{5}$$

From (3b) we see that $\sigma_n \equiv 0$. We can study the synchronous states of (1) by studying the constant solutions of system (4). Each constant solution of (4) corresponds to a one parameter family of equivalent synchronous states of (1); the parameter is the integrating constant for equation (5).

The stability results of this paper are developed by studying the system

$$\dot{y} = -Ay - g(Z) \tag{6a}$$

$$\dot{Z} = By \tag{6b}$$

where $y \in R^k$, $Z \in R^\ell$, A and B are kxk and ℓxk matrices and $g: R^\ell \to R^k$ where $g(0) = 0$. The local stability properties of (6) are determined from the stability properties of the linearized system

$$\dot{y} = -Ay + CZ \tag{7a}$$

$$\dot{Z} = By \tag{7b}$$

where C is the Jacobian matrix for g(Z) evaluated at Z = 0. System (4) can be written in the form of (6) with k=n, ℓ=n-1. A synchronous state of (1) is called stable if it corresponds to an asymptotically stable solution of the linearized system (7). It is not difficult to see that this concept of stability for (1) is consistent with the following definition.

Definition 1

A synchronous state δ^* of (1) is stable if

(i) for every $\varepsilon > 0$ there is a $\gamma > 0$ such that if δ is a solution of (1) with $|\delta_i(0) - \delta_i^*(0)| < \gamma$ and $|\dot{\delta}_i(0) - \dot{\delta}_i^*(0)| < \gamma$ for i=1, ..., n, then $|\delta_i(t) - \delta_i^*(t)| < \varepsilon$ and $|\dot{\delta}_i(t) - \dot{\delta}_i^*(t)| < \varepsilon$ for i=1, ..., n and for all $t \geq 0$; and

(ii) there exists $\gamma > 0$ such that if δ is a solution of (1) with $|\delta_i(0) - \delta_i^*(0)| < \gamma$ and $|\dot{\delta}_i(0) - \dot{\delta}_i^*(0)| < \gamma$ for i=1, ..., n, then there is a constant δ_o such that $\lim_{t\to\infty}|\delta_i(t) - \delta_i^*(t) - \delta_o| = 0$ and

$\lim_{t\to\infty} |\dot{\delta}_i(t) - \dot{\delta}_i^*(t)| = 0$ for i=1, ..., n.

Thus, a synchronous state δ^* is called stable in this paper if it is stable in the usual sense and if all solutions of (1) which are initially close to δ^* will asymptotically approach a synchronous state equivalent to δ^*. Clearly, if δ^* is stable, then all equivalent states are stable.

The stability criteria of this paper are based on the matrices M, H and H_1 as defined below. For system (1) let

$$M = \operatorname{diag}(M_1, \ldots, M_n) \tag{8}$$

For a synchronous state δ^* of (1) let $H = (h_{ij})$ be an nxn matrix whose elements are

$$h_{ij} = \begin{cases} \sum_{\substack{k=1\\k\neq i}}^{n} P_{ik} \cos(\delta_i^* - \delta_k^* - \alpha_{ik}) & \text{if } i=j \\ -P_{ij} \cos(\delta_i^* - \delta_j^* - \alpha_{ij}) & \text{if } i\neq j \end{cases} \tag{9}$$

Since the differences $\delta_i^* - \delta_j^* = \sigma_i - \sigma_j$ are constant, H is a constant matrix. We define H_1 to be the submatrix formed from H by eliminating the last row and column of H.

III. EXISTENCE AND UNIQUENESS RESULTS

The existence of stable synchronous states of (1) is usually predicted by numerical methods. The following theorem is a simple analytic criterion for the existence of a stable synchronous state of (1) which can be applied in the case that transfer conductances are negligible.

Theorem 1

In system (1) suppose that $\alpha_{ij} = 0$ for i, j=1, ..., n, i≠j, that $\sum_{i=1}^{n} P_i = 0$, and that there is an integer k, $1 \le k \le n$ such that

$\sum_{\substack{i=1\\i\neq k}}^{n} P_i^2/P_{ik}^2 < 1$. Then there is a stable synchronous solution δ^* of (1) such that $|\delta_i^*(t) - \delta_j^*(t)| < \pi/2$ for all t and for i, j=1, ..., n.

The hypothesis $\sum_{i=1}^{n} P_i = 0$ can be satisfied for any form of system (1) with $\alpha_{ij} = 0$ by making a simple change of variables.

We can easily see that synchronous states are never unique, for if δ^* is a synchronous state, then we may construct another synchronous state by choosing a constant δ_o and n integers $\{k_1, \ldots, k_n\}$. Then $\delta^{**} = (\delta_1^* + 2k_1\pi + \delta_o, \ldots, \delta_n^* + 2k_n\pi + \delta_o)$ is also a synchronous state. However, we can prove the following results.

Theorem 2

Suppose that δ^* and δ^{**} are two synchronous states of (1) with $|\delta_i^* - \delta_j^* - \alpha_{ij}| < \pi/2$ and $|\delta_i^{**} - \delta_j^{**} - \alpha_{ij}| < \pi/2$ for i, j=1, ..., n, i≠j. Then there is a constant δ_o such that $\delta_i^{**} = \delta_i^* + \delta_o$ for i=1, ..., n; that is, δ^* and δ^{**} are equivalent.

We now consider uniqueness in the case of negligible transfer conductances ($\alpha_{ij} = 0$). From its definition we see that H_1 may be considered as a function of $\sigma^* = (\sigma_1^*, \ldots, \sigma_n^*)$ where the σ_i^* are defined in equation (2). We let $D = \{\sigma^* \epsilon R^n : H_1(\sigma^*)$ is positive definite$\}$.

Theorem 3

In system (1) suppose $\alpha_{ij} = 0$ for i, j=1, ..., n, $i \neq j$. Let T be a convex subset of D. Suppose that σ^* and σ^{**} are two elements of T such that $\delta^* = (\omega_o t + \sigma_1^*, \ldots, \omega_o t + \sigma_n^*)$ and $\delta^{**} = (\omega_o t + \sigma_1^{**}, \ldots, \omega_o t + \sigma_n^{**})$ are synchronous states of (1). Then there is a constant δ_o such that $\delta_i^{**} = \delta_i^* + \delta_o$ for i=1, ..., n; that is, δ^* and δ^{**} are equivalent.

For the case $T = \{\sigma^* : |\sigma_i^* - \sigma_j^*| < \pi/2$, i, j=1, ..., n$\}$ we see that Theorems 3 and 4 are equivalent because $H_1(\sigma^*)$ is positive definite for σ^* in T.

IV. STABILITY RESULTS

There are few results available on the stability of power systems with non-trivial transfer conductances. One paper [4] gives results for the case of uniform damping and small interconnections between machines. The following theorems list local stability results for the case of non-trivial transfer conductances without restriction on the size of the interconnections. Theorems 4(iii) and 6(ii) show that the stability criterion for uniform damping is almost necessary and sufficient; it can fail to apply only in a certain critical case. Theorem 4(iv) gives a criterion which guarantees stability for any amount of uniform damping. The hypotheses can be verified by applying Sturm's theorem [1].

Theorem 4

Let δ^* be a synchronous state of (1). Then δ^* is stable if one of the following criteria is satisfied:

(i) $\delta_i^* - \delta_j^* = 0$ for i, j=1, ..., n

(ii) $\alpha_{ij} = 0$ for i, j=1, ..., n, $i \neq j$ and H_1 is positive definite

(iii) $|\delta_i^* - \delta_j^* - \alpha_{ij}| < \pi/2$ for i, j=1, ..., n, $i \neq j$ and there is a positive constant $a = d_i/M_i$ for i=1, ..., n such that if $\mu + j\nu$ (where $j = \sqrt{-1}$ and μ and ν are real) is a non-zero eigenvalue of $M^{-1}H$, then $\nu^2 - a^2\mu < 0$

(iv) $|\delta_i^* - \delta_j^* - \alpha_{ij}| < \pi/2$ for i, j=1, ..., n, $i \neq j$, there is a positive constant $a = d_i/M_i$ for i=1, ..., n, and all eigenvalues of $M^{-1}H$ are real and non-negative

(v) $|\delta_i^* - \delta_j^* - \alpha_{ij}| < \pi/2$ for i, j=1, ..., n, $i \neq j$ and

$$d_i^2 \geq 2M_i \sum_{\substack{j=1 \\ j \neq i}}^{n} P_{ij} \cos(\delta_i^* - \delta_j^* - \alpha_{ij}) \text{ for } i=1, \ldots, n.$$

By part (i) of the above theorem we see by a continuity argument that if $|\delta_i^* - \delta_j^*|$ is small enough for i, j=1, ..., n, then δ^* is stable.

Part (ii) is a well-known result included for completeness; however, earlier research concluded a weaker form of stability than that of Definition 1. Parts (iii) and (iv) assume uniform damping, an assumption often made. The condition in part (iv) that all eigenvalues of $M^{-1}H$ are real and non-negative may be verified by applying Sturm's theorem to the characteristic polynomial of $M^{-1}H$. (See for example [1]). An algorithm analogous to the Routh-Hurwitz criterion can be developed to implement Sturm's theorem. The condition $|\delta_i - \delta_j - \alpha_{ij}| < \pi/2$ may be removed if it is known that the Jacobian for the system in the form of (4) is non-zero.

We also have the following instability results. (Note that, except in Theorem 5 we have $d_i > 0$.)

Theorem 5

In (1) assume that $d_i = 0$ for i=1, ..., n. If δ^* is a stable synchronous state, then all eigenvalues of $M^{-1}H$ are real and non-negative.

Theorem 6

Let δ^* be an synchronous state of (1). If one of the following criteria is satisfied, then δ^* is not stable:

(i) $\alpha_{ij} = 0$ for i, j=1, ..., n, i≠j and H_1 is not positive semidefinite.

(ii) there is a constant $a = d_i/M_i$ for i=1, ..., n and there is an eigenvalue $\mu + j\nu$ (where $j=\sqrt{-1}$) of $M^{-1}H$ such that $\nu^2 - a^2\mu > 0$.

In fact, if Theorem 6 is satisfied, then part (i) of Definition 1 is violated.

V. EXAMPLES

It is common in transient stability analysis to consider only a synchronous state δ^* with $|\delta_i^* - \delta_j^* - \alpha_{ij}| < \pi/2$ for i, j=1, ..., n, i≠j. It can be shown that a stable synchronous state δ^* with $|\delta_i^* - \delta_j^*| \leq \pi$ may exist even though $|\delta_i^* - \delta_j^*| > \pi/2$ for most of the rotor angle pairs δ_i, δ_j. As the following example shows, the rotor angles may even be 180° out of phase.

Example 1

Let the parameters of (1) be defined n=4, $d_i = 1$, $M_i > 0$, $\alpha_{ij} = 0$ for i, j=1, 2, 3, 4, i≠j and $P_1 = 120\pi + 11\sqrt{3}/2$, $P_2 = 120\pi + \sqrt{3}/2$, $P_3 = 120\pi - \sqrt{3}/2$, $P_4 = 120\pi - 11\sqrt{3}/2$, $P_{12} = P_{23} = P_{34} = 10$, $P_{13} = P_{14} = P_{24} = 1$.

We could imagine this situation arising if machines i and i+1 are adjacent in some sense for i=1, 2, 3 while otherwise the machines are not adjacent. The transfer impedance between adjacent machines is small relative to non-adjacent machines so that the quantity $P_{ij} = |E_iE_j/Z_{ij}|$, $1 \leq i < j \leq 4$, is large for j=i+1 and small for j≠i+1.

In this example there is a synchronous state δ^* with an angular frequency $120\pi = \omega_o$ (frequency 60) and rotor angle functions

$\delta_i^*(t) = (4 - i + 360\,t)\,\pi/3$ for i=1, 2, 3, 4. If we make the transformation $\overline{\delta}_i = \delta_i - \omega_o t$, i=1, 2, 3, 4, then the new system will satisfy Theorem 4 (ii) because in the transformed system we find that $\overline{P}_1 = 11\sqrt{3}/2$, $\overline{P}_2 = \sqrt{3}/2$, $\overline{P}_3 = -\sqrt{3}/2$, $\overline{P}_4 = -\sqrt{3}/2$ and

$$H_1 = \frac{1}{2}\begin{bmatrix} 7 & -10 & 1 \\ -10 & 19 & -10 \\ 1 & -10 & 19 \end{bmatrix}$$

which is positive definite. The synchronous solution in the transformed system has frequency $\omega_o = 0$ and $\overline{\delta}_1^* = \pi$, $\overline{\delta}_2^* = 2\pi/3$, $\overline{\delta}_3^* = \pi/3$, $\overline{\delta}_4^* = 0$. Thus δ^* is stable even though $\delta_1^* - \delta_4^* = \pi$ and other rotor angles are more than 90° out of phase.

It is usually assumed that if a synchronous state δ^* of (1) has the property $|\delta_i^* - \delta_j^* - \alpha_{ij}| < \pi/2$, then it is stable. One author attempted to prove this conjecture [3]. It is certainly true for the case $\alpha_{ij} = 0$ as seen in Theorem 4 (ii), but by Theorem 5 it is probably not always true for the case $\alpha_{ij} \neq 0$ in which H is not symmetric so that eigenvalues of $M^{-1}H$ may be non-real. The following example demonstrates that the conjecture is not true.

Example 2

In system (1) let n=3, $M_1 = M_2 = M_3 = 1$, $0 < d = d_1 = d_2 = d_3 \leq 0.5$, $P_{12} = 3.001149$, $P_{13} = P_{23} = 1.353902$, $P_1 = -3.217683$, $P_2 = 1.376475$, $P_3 = 1.547330$, $\alpha_{12} = 73.62308^\circ$, $\alpha_{13} = 42.87356^\circ$, $\alpha_{23} = -32.87356^\circ$. This system will have a synchronous state with angular frequency $\omega_o = 0^\circ$ and constant rotor angles $\delta_1^* = 30^\circ$, $\delta_2^* = 40^\circ$, $\delta_3^* = -2.87356^\circ$. Angles are written in degrees rather than radians for easy comparison.

We see that $|\delta_i^* - \delta_j^* - \alpha_{ij}| < 90^\circ$ for all i, j=1, 2, 3, i≠j. However, in this case

$$M^{-1}H = H = \frac{1}{3}\begin{bmatrix} 5 & -1 & -4 \\ -4 & 5 & -1 \\ -1 & -4 & 5 \end{bmatrix}$$

which has eigenvalues $\lambda = 0$ and $\lambda = (5 \pm j\sqrt{3})/2$ where $j = \sqrt{-1}$. We note that for $\mu + j\nu = (5 + j\sqrt{3})/2$, $\nu^2 - d^2\mu \geq 3/4 - (1/2)^2\,(5/2) = \frac{1}{8} > 0$. Hence, by Theorem 6 this synchronous state is not stable.

This example demonstrates that any attempt to base a stability criterion on the conjecture mentioned above is bound to fail. The transfer conductances in the example are larger than one would expect in practice, but as the number of machines is increased the chances that the non-symmetric matrix has a non-real eigenvalue will increase. One would expect more realistic examples in higher dimensions.

REFERENCES

[1] L. E. Dickson. New First Course in the Theory of Equations. John Wiley, New York, 1939.

[2] A. A. Fouad. Stability Theory-Criteria for Transient Stability. ERDA Report, Engineering Research Institute, Iowa State University, Ames, 1975.

[3] U. Gudaru. A General Liapunov Function for Multi-Machine Power Systems with Transfer Conductances. Int. J. Contr., 21 (1975), pp. 333-343.

[4] L. B. Jocic, M. Ribbens-Pavella and D. D. Siljak. Multimachine Power Systems: Stability, Decomposition, and Aggregation. IEEE Trans. Automat. Contr., 23 (1978), pp. 325-332.

[5] S. J. Skar. Stability of Multi-Machine Power Systems with Non-Trivial Transfer Conductances. SIAM J. on Applied Math., submitted.

[6] S. J. Skar. Synchronous Solutions of Power Systems: Existence, Uniqueness, and Stability. Quarterly of Applied Math., submitted.

[7] J. L. Willems. Direct Methods for Transient Stability Studies in Power System Analysis. IEEE Trans. Automat. Contr., 16 (1971), pp. 332-341.

DERIVATION OF THE BRAYTON-MOSER EQUATIONS FROM A TOPOLOGICAL MIXED POTENTIAL FUNCTION

F. M. MASSIMO H. G. KWATNY L. Y. BAHAR
Drexel University
32nd and Chestnut Streets
Philadelphia, PA 19104
Mechanical Engineering and Mechanics Department

ABSTRACT

A topological mixed potential function, P_T^*, is defined and shown to be an exact differential. From P_T^*, the Brayton-Moser equations are obtained from topological relationships of the network and the constitutive relationships of the elements. A relationship between P_T^* and Brayton-Moser's mixed potential function, Q, is developed.

1. INTRODUCTION

The Brayton-Moser equations [1], which were introduced in 1964, characterize a general class of nonlinear electrical networks in terms of a "mixed potential function" which is a scalar function of a set of independent capacitor voltages and a set of independent inductor currents. From the mixed potential function, the Brayton-Moser equations yield the differential equations governing the behavior of the electrical networks.

In [2,3], MacFarlane provides a method for selecting a tree and a chord to obtain the maximum number of independent capacitor voltages and the maximum number of independent inductor currents and establishes the topological relationships between the tree elements and the chord elements. MacFarlane then defines the scalar mixed potential function Q via a variational form, δQ, for a nonlinear network without excess elements and subject to restrictions described later in the text. However, in [3] no rigorous proof is given that δQ is a perfect differential and hence his derivation does not guarantee the existence of the mixed potential function. In a letter, Flower [4] shows that δQ is an exact differential for the circuits described in [1]. However, these circuits are not as general as desired. He raises the question of the existence of the mixed potential function for more general circuits. He supplies a partial answer to this question in a subsequent letter [5], in which he proves the existence of Q for a more general class of nonlinear networks, although the proof is tedious and somewhat vague. In this paper, the authors provide a new proof to the effect that the mixed potential function does indeed exist for a very general class of nonlinear networks.

In 1973 Jones and Evans [6] defined the so-called "topological mixed potential function", which is a scalar function of the chord voltages and tree currents. The present paper is based on the notion of a topological mixed potential function, which is defined more naturally, as a scalar function of the chord currents and tree voltages. The topological mixed potential function of Jones and Evans [6] appears only in variational form and no rigorous proof is provided that it is a perfect differential.

This paper contains several extensions to prior work described above. The topological mixed potential function, P_T^* is shown to be an explicit function of the tree voltages and chord currents and this result is obtained by first proving that δP_T^* is an exact differential. With P_T^* adequately defined, the topological relations are obtained which show how

P_T^*, the tree voltages and currents, and the link voltages and currents are related. From these relationships, the Brayton-Moser equations are derived as well as the topological relationships defined in MacFarlane [3]. A rigorous proof is provided to show that δQ,the mixed potential function, is indeed a perfect differential. An example is included which illustrates the removal of certain technical restrictions previously required on the network configurations.

2. NETWORK VARIABLE RELATIONSHIPS

In order to define the topological relationships between the tree elements and chord elements, one must first choose a tree representation. MacFarlane [3] provides a method of selecting a tree and a chord to obtain the maximum number of independent capacitor voltages and the maximum number of independent inductor currents. First, capacitors are chosen for the tree. The only situation in which all of the capacitors are not included in the tree arise from the presence of capacitor-only loops. If this occurs then the capacitors can be divided so that the tree contains an independent set of capacitors and the "excess" capacitors are placed in the chord of the tree. Similarly, inductors are placed in the chord and the only situation in which all of the inductors are not included in the chord is the existence of inductor only cut sets. If a cut set occurs then the inductors can be divided so that the chord contains an independent set of inductors and the "excess" inductors are placed in the tree. This method insures that a maximal number of independent capacitors are included in the tree and the maximal number of independent inductors are included in the chord. MacFarlane calls such a tree a "state tree". The state tree is completed by an arbitrary selection of converter elements, i.e. resistors and/or constant source elements. Two other properties are established by choosing a state tree representation. If a chord capacitor is inserted into a state tree then the only type of loop which can be formed is a loop composed entirely of capacitors. Also, if a chord resistor is inserted into the state tree to form a loop then this loop cannot contain a tree inductor.

For any given tree, a topological relationship exists between the tree and chord currents and between the tree and chord voltages. These relationships are an immediate consequence of Kirchoff's Laws and are as follows:

$$i_t = D\, i_c \tag{1}$$

$$v_c = -D' v_t \tag{2}$$

where

i_t is a vector of tree currents

i_c is a vector of chord currents

v_c is a vector of chord voltages

v_t is a vector of tree voltages

D is the network dynamical transformation matrix defined in MacFarlane [3].

Utilizing the above two properties, equation (1) can be expanded as in [3] as follows

$$\begin{pmatrix} i_{st} \\ i_{nt} \\ i_{et} \\ v_{sc} \\ v_{nc} \\ v_{ec} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & D_{ss} & D_{sn} & D_{sc} \\ 0 & 0 & 0 & D_{ns} & D_{nn} & 0 \\ 0 & 0 & 0 & D_{es} & 0 & 0 \\ -D_{ss}' & -D_{ns}' & -D_{es}' & 0 & 0 & 0 \\ -D_{sn}' & -D_{nn}' & 0 & 0 & 0 & 0 \\ -D_{sc}' & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} v_{st} \\ v_{nt} \\ v_{et} \\ i_{sc} \\ i_{nc} \\ i_{ec} \end{pmatrix} \qquad \begin{matrix} (3a) \\ (3b) \\ (3c) \\ (3d) \\ (3e) \\ (3f) \end{matrix}$$

where

(a) i_{st}, v_{st} are vectors whose components are respectively the currents and voltages of the tree state passive elements, i.e. the tree capacitors;

(b) i_{nt}, v_{nt} are vectors whose components are respectively the currents and voltages of the tree non-state element, i.e. the tree converters;

(c) i_{et}, v_{et} are vectors whose components are respectively the currents and voltages of the tree excess elements, the tree inductors;

(d) v_{sc}, i_{sc} are vectors whose components are respectively the voltages and currents of the chord state elements, i.e. the chord inductors;

(e) v_{nc}, i_{nc} are vectors whose components are respectively the voltages and currents of the chord non-state elements, i.e. the chord converters;

(f) v_{ec}, i_{ec} are vectors whose components are respectively the voltages and currents of the chord excess elements, i.e. the chord capacitors;

(g) D_{ss}, D_{sn}, D_{sc}, D_{ns}, D_{nn}, D_{es} are matrices obtained by the appropriate partitioning of the matrix D corresponding to a state tree.

3. EXISTENCE OF THE TOPOLOGICAL MIXED POTENTIAL FUNCTION, P_T^*

The existence of the topological mixed potential function is proven by showing that δP_T^* is a perfect differential. Also P_T^* will be expressed explicitly for networks which contain no excess elements and excess elements.

The topological mixed potential function is defined as follows:

$$\delta P_T^* \triangleq \langle i_t, \delta v_t\rangle - \langle v_c, \delta i_c\rangle \tag{4}$$

where i_t, v_t, i_c and v_c are defined by equations (1) and (2) and $\langle x,y\rangle$ denotes the inner product of x and y define as $\langle x,y\rangle = x'y$. The following theorem is needed in order to prove that P_T^* exists.

Theorem 1: Let x be an n dimensional column vector and Q(x) an n dimensional row vector-function of x. Q(x) dx is a perfect or exact differential if and only if $\partial Q'/\partial x$ is symmetric, provided the domain of x is simply connected.

The proof of this theorem can be found in Protter and Morrey [7].

By substituting equations (1) and (2) into equation (4) and performing the inner product of (4), the following is obtained,

$$\delta P_T^* = (Di_c)'\delta v_t + (D'v_t)'\delta i_c \tag{5}$$

Equation (5) can be written in the following vector form i.e.

$$\delta P_T^* = [(Di_c)' \mid (D'v_t)'] \left(\begin{array}{c} \delta v_t \\ \hline \delta i_c \end{array}\right) \tag{6}$$

Note that equation (6) is in the form $Q(x)\delta x$ where

$$Q(x) = [(Di_c)' \mid (D'v_t)'] \quad \text{and} \quad x = [v_t \mid i_c]' \tag{7}$$

By the above theorem, it is shown that $\partial Q'/\partial x$ is symmetric thereby proving that $\partial P_T^* \equiv dP_T^*$. Now

$$Q' = \left(\begin{array}{c} Di_c \\ \hline D'v_t \end{array}\right) \quad \text{and}$$

$$\frac{\partial Q'}{\partial x} = \frac{\partial Q'}{\partial \left(\begin{array}{c} v_t \\ \hline i_c \end{array}\right)} = \begin{bmatrix} 0 & D \\ D' & 0 \end{bmatrix} \tag{8}$$

Obviously equation (8) is symmetric and therefore $\partial P_T^* = dP_T^*$. The topological mixed potential function can be explicitly derived as follows

$$P_T^* = \langle v_t, i_t \rangle \tag{9}$$

Equation (9) is obtained as follows:

$$dP_T^* = \langle i_t, dv_t \rangle - \langle v_c, di_c \rangle$$

$$= \langle i_t, dv_t \rangle + \langle v_t, di_t \rangle - \langle v_t, di_t \rangle - \langle v_c, di_c \rangle \tag{10}$$

By using equations (1) and (2) it can be shown that

$$\langle v_t, di_t \rangle = \langle v_t, Ddi_c \rangle = \langle D'v_t, di_c \rangle = -\langle v_c, di_c \rangle.$$

Therefore

$$-\langle v_t, di_t \rangle - \langle v_c, di_c \rangle = 0$$

and upon integrating equation (10) it follows that

$$P_T^* = \langle v_t, i_t \rangle.$$

4. EXISTENCE OF THE MIXED POTENTIAL FUNCTION, Q

The mixed potential function, Q, is defined as in MacFarlane [3] i.e.

$$\delta Q = \langle i_\gamma, \delta v_\gamma \rangle - \langle v_\lambda, \delta i_\lambda \rangle \tag{11}$$

where

γ = network capacitors
λ = network inductors

For consistency, equation (11) can be written as follows:

$$\delta Q = \langle i_{st}, \delta v_{st} \rangle + \langle i_{ec}, \delta v_{ec} \rangle - \langle v_{et}, \delta i_{et} \rangle - \langle v_{sc}, \delta i_{sc} \rangle \tag{12}$$

Note that equation (12) defines Q for excess elements. Utilizing equations (3c) and (3f), equation (12) is rearranged as

$$\delta Q = \langle i_{st} - D_{se} i_{ec}, \delta v_{st} \rangle - \langle v_{sc} + D'_{es} v_{et}, \delta i_{sc} \rangle. \tag{13}$$

By equation (3a)

$$i_{st} - D_{se} i_{ec} = D_{ss} i_{sc} + D_{sn} i_{nc} \tag{14}$$

and by equation (3d)

$$v_{sc} + D'_{es} v_{et} = -D'_{ss} v_{st} - D'_{ns} v_{nt} . \tag{15}$$

For brevity sake, let $A = i_{st} - D_{se} i_{ec}$ and $B = v_{sc} + D'_{es} v_{et}$. Now, δQ can be arranged in a vector form so as to utilize theorem 1. Let

$$Q(x) = [A' \mid -B'] \quad \text{and} \quad x = [v_{st} \mid i_{sc}]'$$

and then

$$\frac{\partial Q'}{\partial x} = \begin{pmatrix} \partial A/\partial v_{st} & \partial A/\partial i_{sc} \\ \partial(-B)/\partial v_{st} & \partial(-B)/\partial i_{sc} \end{pmatrix} = K \tag{16}$$

Now to show that K is symmetric, it must be shown that

(a) $\partial A/\partial v_{st}$ is symmetric

(b) $\partial(-B)/\partial i_{sc}$ is symmetric

(c) $(\partial A/\partial i_{sc})' = \partial(-B)/\partial v_{st}$

It is assumed without loss of generality that the converter elements have the following functional form:

$$v_{nt} = r(i_{nt}) \tag{17}$$

$$i_{nc} = g(v_{nc}) \tag{18}$$

Using equations (3b), (3e), (17) and (18), the relationships of equations (17) and (18) can be represented as the following composite functions:

$$i_{nc} = g(-D'_{sn} v_{st} - D'_{nn} r(D_{ns} i_{sc} + D_{nn} i_{nc})) \tag{19}$$

$$v_{nt} = r(D_{ns} i_{sc} + D_{nn} g(-D'_{sn} v_{st} - D'_{nn} v_{nt})) \tag{20}$$

To show condition (a), take the $\partial A/\partial v_{st}$ using equation (14). This yields

$$\frac{\partial A}{\partial v_{st}} = D_{sn} \frac{\partial i_{nc}}{\partial v_{st}} \tag{21}$$

Now

$$\frac{\partial i_{nc}}{\partial v_{st}} = \frac{\partial g(v_{nc})}{\partial v_{nc}} \frac{\partial v_{nc}}{\partial v_{st}} = \frac{\partial g(v_{nc})}{\partial v_{nc}} \left(-D'_{sn} - D'_{nn} \frac{\partial r(i_{nt})}{\partial i_{nt}} D_{nn} \frac{\partial i_{nc}}{\partial v_{st}} \right) \tag{22}$$

Let $\partial g(v_{nc})/\partial v_{nc} = G$ and $\partial r(i_{nt})/\partial i_{nt} = R$ then solving for $\partial i_{nc}/\partial v_{st}$ in equation (22) yields

$$\frac{\partial i_{nc}}{\partial v_{st}} = -(I + G D'_{nn} R D_{nn})^{-1} G D'_{sn} \tag{23}$$

provided the inverse exists. Therefore,

$$\frac{\partial i_{st}}{\partial v_{st}} = -D_{sn} (I + G D'_{nn} R D_{nn})^{-1} G D'_{sn} \tag{24}$$

The $\partial i_{st}/\partial v_{st}$ is symmetric if it can be shown that

$$(I + G D'_{nn} R D_{nn})^{-1} G = [(I + G D'_{nn} R D_{nn})^{-1} G]' \tag{25}$$

By equations (17) and (18), it is apparent that G and R are diagonal and therefore symmetric. After performing the transpose of the right hand side of equation (25) it becomes

$$G(I + D'_{nn} R D_{nn} G)^{-1} \triangleq G V^{-1} \tag{26}$$

Define the left hand side of (25) as $S^{-1}G$, so that equation (25) is written as

$$S^{-1}G = GV^{-1} \tag{27}$$

Multiply equation (27) on the left by S and on the right by V to yield

$$G(I + D'_{nn} R D_{nn} G) = (I + G D'_{nn} R D_{nn}) G \tag{28}$$

It is now obvious that equation (28) is an identity therefore $\partial i_{st}/\partial v_{st}$ is symmetric. The proof of condition (b) is similar to that of (a) and will therefore be excluded from this paper. The last condition (c) is also relatively straightforward to prove. After taking partial derivatives of equation (14) and some substitutions, it can be shown that

$$\frac{\partial i_{st}}{\partial i_{sc}} = D_{ss} - D_{sn} (I + G D'_{nn} R D_{nn})^{-1} G D'_{nn} R D_{ns} . \tag{29}$$

Therefore,

$$\left[\frac{\partial i_{st}}{\partial i_{sc}}\right]' = D'_{ss} - D'_{ns} R D_{nn} G(I + D'_{nn} R D_{nn} G)^{-1} D'_{sn} \tag{30}$$

Similarly, $\partial(-v_{sc})/\partial v_{st}$ is computed by taking the partial derivation of equation (15) and some substitutions to yield

$$\frac{\partial(-v_{sc})}{\partial v_{st}} = D'_{ss} - D'_{ns} (I + R D_{nn} G D'_{nn})^{-1} R D_{nn} G D'_{sn} \tag{31}$$

By comparison of equations (30) and (31) condition (c) can be proven if it can be shown that

$$RD_{nn} G(I + D'_{nn} R D_{nn} G)^{-1} = (I + R D_{nn} G D'_{nn})^{-1} R D_{nn} G \tag{32}$$

Proceeding as before, that is to eliminate the inverses of equation (32), the identity becomes apparent. Therefore, condition (c) is satisfied and K is symmetric. By theorem 1 this implies that δQ is an exact differential.

MacFarlane [3] derives explicitly the Q function for networks with no excess elements. The derivation is not repeated in this paper. However, if one follows MacFarlane's derivation, a slight mistake occurs which alters the final derivation of Q. The correct Q function is given as

$$Q = v'_{st} D_{ss} i_{sc} - v'_{nt} D_{nn} i_{nc} + G_{tree} - J_{chord} \tag{33}$$

where G_{tree} and J_{chord} are the total content of the tree converters and the total co-content of the chord converters. For networks which contain excess elements,

$$Q = \langle v_{nc}, i_{nc} \rangle + \langle v_{st}, i_{st} \rangle + \langle v_{ec}, i_{ec} \rangle + G_{tree} - J_{chord} \tag{34}$$

This is not derived in MacFarlane. However, if one follows MacFarlane's derivation for the non-excess element Q, the excess element Q can be easily verified.

5. DEVELOPMENT OF BRAYTON-MOSER EQUATION FROM P^*_T

Before the Brayton-Moser equations are developed from the topological mixed potential function, certain topological equations must be derived. Note, that these relationships are strictly topological and therefore independent of the elements of the network. However to establish Brayton-

Moser's equations, it is necessary to involve the constitutive relations of the elements.

In section 3, it was proven that dP_T^* is an exact differential. Also, it was established that P_T^* is a function only of the tree voltages, v_t, and the chord currents, i_c. Therefore,

$$dP_T^* = \frac{\partial P_T^*}{\partial v_t} dv_t + \frac{\partial P_T^*}{\partial i_c} di_c \tag{35}$$

However, the definition of dP_T^* is

$$dP_T^* = \langle i_t, dv_t \rangle - \langle v_c, di_c \rangle = i_t' dv_t - v_c' di_c \tag{36}$$

Comparison between equations (35) and (36) yields

$$\frac{\partial P_T^*}{\partial v_t} = i_t' \quad \text{and} \quad \frac{\partial P_T^*}{\partial i_c} = -v_c' \ . \tag{37}$$

These are the topological relationships needed to establish the Brayton-Moser equations.

First, the Brayton-Moser equations are derived with networks containing no excess elements. P_T^* can be written in terms of Q as

$$P_T^*(v_{st}, v_{nt}, i_{sc}, i_{nc})' = Q(v_{st}, i_{sc}) + \int i_{nt}' dv_{nt} - \int v_{nc}' di_{nc} \tag{38}$$

Differentiating equation (38) and comparing terms yields the following relationships

$$\frac{\partial P_T^*}{\partial v_{st}} = \frac{\partial Q}{\partial v_{st}} \tag{39a}$$

$$\frac{\partial P_T^*}{\partial i_{sc}} = \frac{\partial Q}{\partial i_{sc}} \tag{39b}$$

$$\frac{\partial P_T^*}{\partial v_{nt}} = i_{nt}' \tag{39c}$$

$$\frac{\partial P_T^*}{\partial i_{nc}} = v_{nc}' \tag{39d}$$

However comparison with equations (37), (39a) and (39b) it follows that

$$\frac{\partial Q}{\partial v_{st}} = i_{st}' \tag{40}$$

$$\frac{\partial Q}{\partial i_{sc}} = -v_{sc}' \tag{41}$$

The comparison of equations (37), (39c) and (39d) leads to the topological identities of equation (3). Since the elements of i_{st} are capacitors, then the constitutive equation of i_{st} is

$$i_{st} = C_{st} \frac{dv_{st}}{dt} \ . \tag{42}$$

Also, the elements of v_{sc} are inductors and they have the constitutive equation of

$$v_{sc} = L_{sc} \frac{di_{sc}}{dt} \quad . \tag{43}$$

Therefore equating (40) to (42) and (41) to (43), the Brayton-Moser equations are established i.e.

$$\left(\frac{\partial Q}{\partial v_{st}}\right)' = C_{st} \frac{dv_{st}}{dt} \tag{44}$$

$$-\left(\frac{\partial Q}{\partial i_{sc}}\right)' = L_{sc} \frac{di_{sc}}{dt} \tag{45}$$

Second, the preceding analysis can be extended to networks containing excess elments. Again P^*_T is written in terms of Q as

$$\begin{aligned} P^*_T(v_{st},v_{et},v_{nt},i_{sc},i_{ec},i_{nc}) &= Q(v_{st},i_{st}) + <v_{et},i_{et}> - <v_{ec},i_{ec}> \\ &+ \int <i_{nt},dv_{nt}> - \int <v_{nc},di_{nc}> \end{aligned} \tag{46}$$

Utilizing equation (3) and differentiating equation (46) yields

$$\begin{aligned} dP^*_T &= \left(\frac{\partial Q}{\partial v_{st}} + i'_{ec} D'_{se}\right) dv_{st} + (D_{es} i_{sc}) dv_{et} + (D_{ns} i_{sc} + D_{nn} i_{nc}) dv_{nt} \\ &+ \left(\frac{\partial Q}{\partial i_{sc}} + v'_{et} D_{es}\right) di_{sc} + (v_{st} D_{se}) di_{ec} + (D'_{sn} v_{st} + D'_{nn} v_{nt})' \, di_{nc} \, . \end{aligned} \tag{47}$$

Comparison of equation (47) with the extension of equation (35) yields

$$\frac{\partial P^*_T}{\partial v_{st}} = \frac{\partial Q}{\partial v_{st}} + i'_{ec} D'_{se} \tag{48a}$$

$$\frac{\partial P^*_T}{\partial i_{sc}} = \frac{\partial Q}{\partial i_{sc}} + v'_{et} D_{es} \tag{48b}$$

$$\frac{\partial P^*_T}{\partial v_{et}} = (D_{es} i_{sc})' \tag{48c}$$

$$\frac{\partial P^*_T}{\partial v_{nt}} = (D_{ns} i_{sc} + D_{nn} i_{nc})' \tag{48d}$$

$$\frac{\partial P^*_T}{\partial i_{ec}} = (v'_{st} D_{se}) \tag{48e}$$

$$\frac{\partial P^*_T}{\partial i_{nc}} = (D'_{sn} v_{st} + D'_{nn} v_{nt})' \tag{48f}$$

Now comparison with equation (37), (48a) and (48b) it follows that

$$\frac{\partial Q}{\partial v_{st}} + i'_{ec} D'_{se} = i'_{st} \tag{49}$$

$$\frac{\partial Q}{\partial i_{sc}} + v'_{et} D_{es} = -v'_{sc} \tag{50}$$

As before, comparison of equations (37) and (48c-f) leads to the topological identities of equation (3). As for equations (49) and (50), they are again the extended Brayton-Moser equation. Recall that i_{ec} and i_{st} are capacitor currents and therefore follow the constitutive equations of (42). Similarly, v_{ct} and v_{sc} are inductor voltages and therefore follow the constitutive equation of (43). Equations (49) and (50) can be written as

$$\left(\frac{\partial Q}{\partial v_{st}}\right)' = C_{st}\frac{dv_{st}}{dt} - D_{se}C_{ec}\frac{dv_{ec}}{dt} \tag{51}$$

$$-\left(\frac{\partial Q}{\partial i_{sc}}\right)' = L_{sc}\frac{di_{sc}}{dt} + D'_{es}L_{et}\frac{di_{et}}{dt} \tag{52}$$

And finally utilizing equations (3c) and (3f), Brayton-Moser's equations are developed i.e.

$$\left(\frac{\partial Q}{\partial v_{st}}\right)' = (C_{st} + D_{se}C_{ec}D'_{se})\frac{dv_{st}}{dt} \tag{53}$$

$$-\left(\frac{\partial Q}{\partial i_{sc}}\right)' = (L_{sc} + D'_{es}L_{et}D_{es})\frac{di_{sc}}{dt} \tag{54}$$

6. EXAMPLE

An example is included in this paper to illustrate a network which is avoided in most examples. The submatrice D_{nn} is always equal to zero in examples found in all the references in this paper. Even in the theory development of [3], $D_{nn} = 0$. It is not necessary to exclude this case. From Figures 1 and 2, the following topological relationship can be derived:

$$\begin{pmatrix} i_c \\ i_{R1} \\ i_{R3} \\ i_e \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} i_L \\ i_{R2} \\ i_j \end{pmatrix} \tag{55}$$

Note that

$$D_{nn} = \begin{pmatrix} 0 & 0 \\ 1 & 1 \\ 0 & 0 \end{pmatrix} \neq 0.$$

Now assume linear resistive elements such that

$$v_{R1} = R_1 i_{R1}$$
$$v_{R3} = R_3 i_{R3}$$
$$i_{R2} = G_2 v_{R2}$$

The mixed potential function Q is

$$Q = v_c i_L + \frac{1}{2}R_1 i_L^2 + \frac{1}{2}R_3 i_{R3}^2 + e i_L - \frac{1}{2}G_2 v_{R2}^2 - j v_{R2} - R_3 G_2 i_{R3} v_{R2} - R_3 i_{Re} j \tag{56}$$

where from the topology and the constitutive relationships

$$v_{R2} = -(1 + R_3 G_2)^{-1}(v_c + R_3 j)$$
$$i_{R3} = -(1 + R_3 G_2)^{-1}(G_2 v_c - j)$$

As can be seen from equation (56), Q is a function of v_c and i_L only.

Now equations (40) and (41) yields one voltage loop equation and one nodal equation i.e.

$$v_L = -(v_c + R_1 i_L + e)$$

and

$$i_c = i_L + i_{R2} + j\ .$$

7. CONCLUSIONS

The emphasis of this paper is to show how the Brayton-Moser equations can be obtained from topological relationships of the network via the topological mixed potential function. In order to obtain this result, it became necessary to prove that the topological mixed potential function exists. This paper also provides a formal proof of the existence of the mixed potential function, which is not only an important result in itself but also a necessary step in the derivation of the Brayton-Moser equations. An explicit relationship between the topological mixed potential function and the mixed potential function is established. This paper includes an unique example in which none of the submatrices of the network dynamical transformation matrix vanish.

REFERENCES

[1] R. K. Brayton and J. K. Moser, "A Theory of Nonlinear Networks I, II" Quarterly of Applied Mathematics, Vol. 12, Nos. 1 and 2. pp. 1-33 and pp. 81-104, April, 1964 and July 1964.

[2] A. G. J. MacFarlane, "Formulation of the State-Space Equations for Non-Linear Networks", Int. J. Control, Vol. 5, No. 2, 1967 pp. 145-161.

[3] A. G. J. MacFarlane, "An Integral Invariant Formulation of a Canonical Equation Set for Non-Linear Electrical Networks", Int. J. Control Vol. 11, No. 3, 1970, pp. 449-470.

[4] J. O. Flower, "The Topology of the Mixed Potential Function", I.E.E.E. Proceedings, Vol. 56, No. 10, Oct. 1968, pp. 1721-1722.

[5] J. O. Flower, "The Existence of the Mixed Potential Function", I.E.E.E. Proceedings, Vol. 56, No. 12, Oct. 1968, pp. 1735-1736.

[6] D. L. Jones and F. J. Evans, "Variational Analysis of Electrical Networks", Journal of the Franklin Inst., Vol. 295, No. 1, Jan. 1973, pp. 9-23.

[7] M. H. Protter and C. B. Morrey, A First Course in Real Analysis, Springer-Verlag, New York, 1977, pp. 431-433.

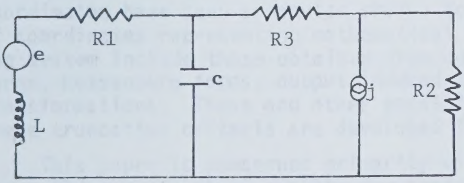


Figure 1

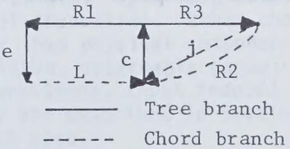


Figure 2

DESIGN OF TIME-FREQUENCY CODED SIGNALS FOR SPREAD-SPECTRUM COMMUNICATIONS

GEORGE R. COOPER
Purdue University
W. Lafayette, IN 47907

SUMMARY

A spread-spectrum communication system intended to provide for unscheduled access by a large number of potential users must have an equally large set of signals with desirable correlation properties. The set of time-frequency coded signals meets these requirements to a reasonable degree. These signals consist of short, frequency-hopped waveforms in which the time-frequency sequences are selected by means of an algorithm that promotes the desirable properties.

The existence of a uniform lower bound on the crosscorrelation function envelope is demonstrated and shown to be inversely proportional to the square root of the time-bandwidth product of the signal set for signals that have the same energy spectrum. An identical result is obtained in the mean for signals whose discrete frequencies are uniformly distributed over the same bandwidth.

A procedure for designing time-limited frequency-hopped signals that possess a "one-coincidence" property is described and examples displayed. Computed crosscorrelation function envelopes demonstrate that these signals compare well with the uniform lower bound on the average and have correlation peaks that exceed the bound by a relatively small amount. A method for bi-phase modulating these signals is also discussed.

The number of signals contained in the time-frequency coded signal set is derived and shown to be proportional to the square of the time-bandwidth product of the set and inversely proportional to the square of the product of chip duration and minimum frequency shift. This conclusion leads to paradox that is still not completely resolved, although some steps toward its resolution have been taken and are discussed. A bound on the maximum number of signals with good crosscorrelation properties is hypothesized.

When time-frequency coded signals are used in channels in which the coherence bandwidth is small compared to the signal set bandwidth, it is necessary to employ differential message modulation. This mode of operation is typical of mobile communication systems, and the application of this signal set in such a system is briefly described. A differentially-coherent matched filter receiver structure is appropriate for detecting such signals in an interference-limited environment. The performance of this receiver in the presence of many interfering signals has been determined by computer simulation and the results are presented.

MODAL COST ANALYSIS IN MODEL AND CONTROLLER REDUCTIONS

ROBERT E. SKELTON
School of Aeronautics & Astronautics
Purdue University
West Lafayette, In. 47907

ABSTRACT

Many model reduction schemes can be interpreted as performing the following sequence of tasks; (1) selecting a set of coordinates or subsystem components which are subject to truncation, and (2) selecting a truncation criterion. A Model Error Index (MEI) can be used to evaluate reduced models when compared to a more complete "evaluation" model. The MEI is based upon a quadratic measure of the difference in outputs of the two models. The quadratic cost for the evaluation model may be decomposed into separate terms involving each subsystem (component contribution). These "component costs" are used as a truncation criterion (2), and the main objective of the paper is to provide and analyze explicit component cost formulas for various definitions of "components" implied by a) output decoupling transformations, b) disturbance decoupling transformations, c) component decoupling transformations, and d) cost decoupling transformations.

1.0 INTRODUCTION

Many model reduction schemes can be interpreted as the following sequence of three tasks 1.1, 1.2, and 1.3. The objectives of this paper are described in this setting.

1.1 Selection of Suitable Coordinates

One must first make the selection of a set of coordinates in which "coordinate truncation" will be performed. Label these coordinates x_i, i=1, 2, ... n and write the system "evaluation model"

$$\dot{x}_i = \sum_{j=1}^{n} A_{ij} x_j + D_i w, \quad i=1,2, \ldots n, \quad y = \sum_{j=1}^{n} C_j x_j \tag{1.1}$$

The coordinates $x_i(t)$ might represent the state of the ith physical subsystem. Under the rules of the present game some subsystems (which we will call system "components") must be deleted from the model (1.1) for any one of a variety of reasons, including: (1) simplified analysis of a large composite system (1.1), (2a) the design of simple centralized controls, and (2b) the design of decentralized controls. To achieve the objectives (1) or (2) it might not be necessary (nor desirable) to restrict the choice of coordinates $x_i(t)$ to represent *physical* components. For example, $x_i(t)$ represents a *mathematical* component of the system if (1.1) is in Jordan form $A_{ij} = J_i \delta_{ij}$. In this case the component x_i is a "modal" coordinate. Open loop modal coordinates have been a popular choice for model truncation. Other choices of coordinates representing mathematical rather than physical components of the system include those obtained from controllable, observable canonical forms, Hessenberg forms, output induced transformations, input induced transformations. These and other possibilities are described in Section 5, where truncation criteria are developed for each case.

This paper is concerned primarily with task (1) above, and a companion paper [1] extends these results for task (2). By the words "model truncation", we will mean the delection of certain components (say the last n-r) from (1.1) to produce the truncated model

$$\dot{x}_i = \sum_{j=1}^{r} A_{ij}x_j + D_i w + e_{x_i}, \quad i=1,2, \ldots r, \quad y = \sum_{j=1}^{r} C_j x_j + e_y \tag{1.2}$$

where the "state model error vectors" $e_{x_i} \triangleq \sum_{j=r+1}^{n} A_{ij}x_j$ (representing the component connections) and the "output model error vector" $e_y \triangleq \sum_{j=r+1}^{n} C_j x_j$, are ignored in the reduced model (obtained from (1.2) by setting e_{x_i}, e_y to zero). The disturbance w will be characterized in section 2.

1.2 Selection of a Truncation Criterion

After the above decomposition of the system into suitable subsystems (components), which components of (1.1) should be truncated? The principal contribution of this paper is the description and analysis of a component truncation criterion called the "component cost", defined in section 3. The component cost V_i is the contribution that component $x_i(t)$ makes in the total cost

$$V = \sum_{i=1}^{n} V_i \quad , \quad V \triangleq \lim_{t\to\infty} \mathcal{E} y^T y \tag{1.3}$$

By ranking the components x_i of (1.1) in order of their component costs, $V_1 \geq V_2 \geq \cdot\cdot V_n$, and truncating those components with the smallest component costs, the reduced model retains those components which make the largest cost contributions. When the choice of components x_i are the "modal" components, then the V_i are called the *modal costs*.

1.3 Evaluation of The Reduced Model

A Model Error Index (MEI) is defined by

$$\text{MEI} = \frac{\delta V}{V} , \quad \delta V \triangleq \lim_{t\to\infty} \mathcal{E}(y-y_R)^T(y-y_R) \tag{1.4}$$

where V is given by (1.3) and Y_R is

$$Y_R = \sum_{j=1}^{r} C_j \hat{x}_j \tag{1.5}$$

defined by (1.2) with e_y ignored ($\hat{x}_j$, j=1, ... r satisfies (1.2) with e_{x_j} ignored). Properties of this index are given in Section 5 for each choice of component definitions.

2.0 DISTURBANCE MODELS

In those control problems in which the control affectors or "actuators", are defined and located a priori, the matrices B_i are known in the reference model, or "evaluation model" of the system.

$$\dot{x}_i = \sum_{j=1}^{n} A_{ij}x_j + D_i w, \quad i=1, 2, \ldots n \tag{2.1a}$$

$$y = \sum_{j=1}^{n} C_j x_j \ , \qquad D_i w(t) \triangleq B_i u(t) + D_i' w'(t) \tag{2.1b}$$

$$V = \lim_{t\to\infty} E \ \| y \|^2 \ , \quad \| y \|^2 \triangleq y^* y \ , \quad (* = \text{conjugate transpose}) \tag{2.1c}$$

Since this paper is dedicated to the examination of *open-loop* component costs, we do not presume to have any time or state-correlated information about u(t). Let $u_c(t)$ and $u_w(t)$ represent the control and disturbance inputs, respectively, emanating from the actuators so that $u(t) = u_c(t) + u_w(t)$ where $u_w(t)$ is a zero mean white noise with covariance U and the mean value function $u_c(t)$ is assumed zero for this paper. (The companion paper [1] treats model reductions with state dependent inputs). If polynomial-type mean value functions are presumed for u(t), instead of zero mean, then the reader may refer to the least squares model reductions for polynomial inputs treated in [2]. If there is any correlation between $u_w(t)$ and w(t) denote it by $E[u_w(t)\ w^*(t)] = C$. Denote the covariance of w' as $\mathcal{Q}'$. By noting that

$$D\mathcal{Q}D^* \triangleq E[Dw(t)w^*(t)D^*] = BUB^*+BCD'+D'C^*B^*+D'\mathcal{Q}'D'^*, \ Dw(t) \triangleq Bu(t)+D'w'(t) \tag{2.2}$$

the value of V in (2.1c) is readily described by

$$V = E[w(t)^* D^* K D w(t)] = \operatorname{tr} K D\mathcal{Q}D^* \ , \qquad 0 = KA + A^*K + C^*C \tag{2.3}$$

where $\mathcal{Q}$ is defined by (2.2). The matrices A, C, and D follow from the compact form of (2.1),

$$\dot{x} = Ax + Dw, \qquad y = Cx. \tag{2.1}$$

The disturbances propogating through the actuators are the natural environmental disturbances including aerodynamics, and moving parts on the structure. Uncertainties in these effects are described by w'(t).

3.0 COMPONENT COST ANALYSIS (CCA) OF LINEAR SYSTEMS

In some linear problems the excitation is assumed through the initial condition whose statistics are given. In other problems, it is more reasonable to assume excitation through random disturbances whose statistics are given. So that the results of the paper may be utilized in either choice of problems, we cite the following equivalence. Under the condition $P_o = \mathcal{Q}$ it may be shown that the problems ZSR and ZIR defined by

$$\text{Problem ZSR} \begin{cases} \dot{x} = Ax + w, \ E[ww^*] = \mathcal{Q} \\ V_{ZSR} = \lim\limits_{t\to\infty} E[x^*Qx] \end{cases} \tag{3.1}$$

$$\text{Problem ZIR} \begin{cases} \dot{x} = Ax \ , \quad E[x(o)x^*(o)] = P_o \\ V_{ZIR} = E \displaystyle\int_0^\infty x^*Qx\,dt \end{cases} \tag{3.2}$$

are equivalent in the sense that $V_{ZSR} = V_{ZIR} = V = \operatorname{tr} K\mathcal{Q}$ where K satisfies (2.3). This paper presents results for Problems ZSR. Setting $D\mathcal{Q}D^* = P_o$ everywhere will give the equivalent results for Problem ZIR. The component costs V_i must represent the total contribution in V due to the excitation $w_i(t)$ or the initial condition $x_i(o)$. All the terms involving $w_i(t)$ (or $x_i(o)$) may be written

$$V_i = \frac{1}{2}\sum_{j=1}^{n}(V_{ij} + V_{ji}) = \sum_{j=1}^{n} Re\, V_{ij} \tag{3.3a}$$

where for the w(t) excitation of problem ZSR,

$$V_{ij} = E[w^*(t)D_i^*K_{ij}D_jw(t)] = trK_{ij}D_j\mathcal{Q}D_i^* \tag{3.3b}$$

The component costs V_i, i=1,2,... n defined by (3.3) are real numbers and have the property that the sum of all component costs equals the total cost $V = \sum_{i=1}^{n} V_i = trKD\mathcal{Q}D^*$. If the componets x_i of (2.1) are arranged according to the order of the component costs

$$V_1 \geq V_2 \geq V_3 \geq \cdots \geq V_n \tag{3.4}$$

and if the n-r components x_i with the smallest component costs are truncated, then the evaluation model and the reduced model may be written respectively as (3.5) and (3.6)

$$\begin{pmatrix} \dot{x}_R \\ \dot{x}_T \end{pmatrix} = \begin{bmatrix} A_R & A_{RT} \\ A_{TR} & A_T \end{bmatrix}\begin{pmatrix} x_R \\ x_T \end{pmatrix} + \begin{bmatrix} D_R \\ D_T \end{bmatrix} w\ , \qquad D\mathcal{Q}D^* = \left[\begin{array}{c|c} D_R\mathcal{Q}D_R^* & D_R\mathcal{Q}D_T^* \\ \hline D_T\mathcal{Q}D_R^* & D_T\mathcal{Q}D_T^* \end{array}\right] \tag{3.5a}$$

$$y = [C_R \quad C_T]\begin{pmatrix} x_R \\ x_T \end{pmatrix} = C\,x \tag{3.5b}$$

$$\dot{\hat{x}}_R = A_R\,\hat{x}_R + D_Rw\ , \qquad y_R = C_R\,\hat{x}_R \tag{3.6}$$

where $x^T_R = (x_1^T, x^T_2, \ldots x_r^T)$ and $x_T^T = (x_{r+1}^T, \ldots x_n^T)$, $C_R = [C_1, C_2, \ldots C_R]$, $C_T = [C_{r+1}, \ldots C_n]$ and the matrices A_R, A_{RT}, A_{TR}, A_T, $\mathcal{Q}_R$, $\mathcal{Q}_{RT}$, and $\mathcal{Q}_T$ follow from (2.1). In the next section the quality of the reduced model (3.6) is evaluated.

4.0 A MODEL ERROR INDEX (MEI)

The quality of any reduced model (3.6) (regardless of how (3.6) may have been derived) may be measured by the MEI in (1.4). y is the output of the evaluation model (2.1), or equivalently (3.5), and y_R is the output of the reduced model (3.6). δV may be determined by defining a model whose output is $(y-y_R)$ and employing the value formula similar to (2.3). Such a model is

$$\dot{x}' = \begin{pmatrix} \dot{x} \\ \dot{\tilde{x}}_R \end{pmatrix} = \begin{bmatrix} A & 0 \\ A & A_R \end{bmatrix}\begin{pmatrix} x \\ \tilde{x}_R \end{pmatrix} + \begin{bmatrix} D \\ 0 \end{bmatrix} w = A'x' + D'w \tag{4.1}$$

$$(y-y_R) = [C \quad C_R]\begin{pmatrix} x \\ \tilde{x}_R \end{pmatrix} = C'x' \tag{4.2a}$$

$$A = [0 \;\; A_{RT}] \;, \quad C = [0 \;\; C_T] \;, \quad \tilde{x}_R \overset{\Delta}{=} x_R - \hat{x}_R \tag{4.2b}$$

The cost associated with model (4.2) is

$$\delta V = \lim_{t\to\infty} E\| y-y_R\|^2 = \mathrm{tr}[K'D'\mathcal{Q}D'^T] = \mathrm{tr}\tilde{K}D\mathcal{Q}D^* \tag{4.3a}$$

where D' is defined in (4.1) and K', $\tilde{K}$ are defined by

$$0 = K'A' + A'^*K' + C'^*C' \;, \quad K' = \left[\begin{array}{c|c} \tilde{K} & K \\ \hline K^* & \underset{\sim}{K} \end{array}\right], \quad K^* = [K_1^*, \;\; K_2^*] \tag{4.3b}$$

Only $\tilde{K}$ is needed in (4.3a)(and not the entire K' matrix). Examination of the partition of (4.3b) leads to these *reduced* computations for $\tilde{K}$ (compute in the listed order),

$$0 = \underset{\sim}{K}A_R + A_R^*\underset{\sim}{K} + C_R^*C_R \tag{4.4a}$$

$$0 = KA_R + A^*K + Q_1 \;, \quad Q_1 \overset{\Delta}{=} \begin{bmatrix} 0 \\ C_T^*C_R + A_{RT}^*\underset{\sim}{K} \end{bmatrix} \tag{4.4b}$$

$$0 = \tilde{K}A + A^*\tilde{K} + Q_2 \;, \quad Q_2 \overset{\Delta}{=} \left[\begin{array}{c|c} 0 & K_1 A_{RT} \\ \hline A_{RT}^*K_1^* & K_2 A_{RT} + A_{RT}^*K_2^* + C_T^*C_T \end{array}\right] \tag{4.4c}$$

Solution of the three smaller linear equations (4.4a) - (4.4c) is more efficient than the solution of the single large linear equation (4.3b).

We wish now to have a convenient expression for calculation of the MEI defined by (1.4). It is convenient to define $\Delta K \overset{\Delta}{=} K-\tilde{K}$ and subtract (4.4c) from (2.3) to get,

$$0 = \Delta KA + A^*\Delta K + (C^*C - Q_2) \tag{4.5}$$

Then the costs (2.3), (4.3a) become

$$V = V_R + \delta V = \mathrm{tr}KD\mathcal{Q}D^* \;, \quad \delta V = \mathrm{tr}\;\tilde{K}D\mathcal{Q}D^*, \quad V_R = \mathrm{tr}\Delta KD\mathcal{Q}D^* \tag{4.6}$$

and the MEI has the value

$$\mathrm{MEI} = \frac{\delta V}{V} = \mathrm{tr}\;\tilde{K}D\mathcal{Q}D^*\;[\mathrm{tr}\;KD\mathcal{Q}D^*]^{-1} \tag{4.7}$$

the MEI is zero for a "perfectly complete" model in the sense $\delta V = 0$. The results of this section are summarized in the following theorem.

Theorem 4.1:

The quality of the model

$$\dot{\hat{x}}_R = A_R\hat{x}_R + D_Rw \;, \qquad y_R = C_R\hat{x}_R \tag{4.8}$$

with respect to the model

$$\dot{x} = \begin{bmatrix} A_R & A_{RT} \\ A_{TR} & A_T \end{bmatrix} x + \begin{bmatrix} D_R \\ D_T \end{bmatrix} w \;, \quad y = [C_R \;\; C_T]\, x \tag{4.9}$$

$$E[w(t)w^*(\tau)] = \mathcal{Q}\delta(t-\tau)$$

is characterized by the Model Error Index (MEI), (4.7). *The component cost is*

$$V_i = \sum_{j=1}^{n} Re\, V_{ij}\ , \qquad V_{ij} = \text{tr}\, K_{ij}\, D_j \mathcal{Q} D_i^* \tag{4.10}$$

5.0 ON THE CHOICE OF COMPONENTS FOR COMPONENT COST ANALYSIS

This section chooses various coordinate transformations on the original system so that in the truncation coordinates (4.9) some of the coupling parameters (A_{RT}, C_T, or $\mathcal{Q}_{RT}$) are zero. The only difference between the model reduction schemes to be discussed is the initial choice of coordinates made prior to truncation. (a choice is not always available if *physical* components are to be deleted). If a choice is made, it is often made so as to *decouple* those entities we truncate from those we retain. Ideally, if this can be done, then no penalty is paid for truncation and the model error index is 0. Since this paper and the MEI of (4.7) is concerned only with preserving output performance y(t), the only components which can be truncated with impunity are those which are unobservable in y(t). We will now select components to "decouple" something. The first choice will be for *disturbance* decoupling ($\mathcal{Q}_{RT} = 0$), then component *cost* decoupling (e.g. $K_{ij} = 0$ for $i \neq j$), *output* decoupling ($C_T = 0$), and finally, *component* decoupling ($A_{RT} = 0$). The Jordan form of A is the extreme case of component decoupling, and this case will be treated under Modal Cost Analysis.

Let the original coordinates be described by $\dot{x}^o = A^o x^o + D^o w$, $y = C^o x^o$. A coordinate transformation $x = Tx^o$ will now be presumed applied to the original coordinates so that (4.9) is the transformed system and T is chosen to enforce $A_{RT} = 0$, for *component* decoupling, or $C_T = 0$ for *output* decoupling, or $\mathcal{Q}_{RT} = 0$ for *disturbance* decoupling, or $V_{ij} = V_{ii}\,\delta_{ij}$ for *cost* decoupling, where

$$A = \begin{bmatrix} A_R & A_{RT} \\ A_{TR} & A_T \end{bmatrix} = TA^oT^{-1}\ , \quad D = \begin{bmatrix} D_R \\ D_T \end{bmatrix} = TD^o\ , \; C = [C_R\ C_T] = C^oT^{-1} \tag{5.1}$$

$$D\mathcal{Q}D^* = \begin{bmatrix} D_R\mathcal{Q}D_R^* & D_R\mathcal{Q}D_T^* \\ D_T\mathcal{Q}D_R^* & D_T\mathcal{Q}D_T^* \end{bmatrix} = TD^o\mathcal{Q}D^{o*}T^* \tag{5.2}$$

5.1 Disturbance Decoupled Components ($\mathcal{Q}_{RT} = 0$)

Let T in (5.1b) have the property $D_R\mathcal{Q}D_T^* = 0$. Then the excitation of the retained components x_R is uncorrelated with the excitation of the truncated components x_T, and the MEI and component costs of theorem 4.1 reduces to

Corollary 1: Uncorrelated Disturbances

If $D_R\mathcal{Q}D_T^* = 0$ *then the model error index for model* (4.8) *is* (4.7) *where*

$$\begin{aligned} \delta V &= \text{tr}(\tilde{K}_R D_R\mathcal{Q}D_R^* + \tilde{K}_T D_T\mathcal{Q}D_T^*) \\ V_R &= \text{tr}(\Delta K_R D_R\mathcal{Q}D_R^* + \Delta K_T D_T\mathcal{Q}D_T^*) \end{aligned} \qquad \Delta K = \begin{bmatrix} \Delta K_R & \Delta K_{RT} \\ \Delta K_{RT}^* & \Delta K_T \end{bmatrix}\ , \quad \tilde{K} = \begin{bmatrix} \tilde{K}_R & \tilde{K}_{RT} \\ \tilde{K}_{RT}^* & \tilde{K}_T \end{bmatrix} \tag{5.3}$$

where $\tilde{K}$ *and* ΔK *satisfy* (4.4c) *and* (4.5), *respectively. The component cost is given by* (4.10) *where* K *satisfies* (2.3), *and where* $D_j\mathcal{Q}D_i^* = 0$ *for* $i = r+1, \ldots n$, $j = 1, \ldots r$.

In the limiting case where *all* component disturbances $D_i w(t)$, i=1,... n are uncorrelated then

$$DQD^* = \text{block diag.}\{\dots D_i Q D_i^* \dots\},\ i=1,\ 2,\ \dots\ n \tag{5.4}$$

and

$$\delta V = \mathrm{tr}\sum_{i=1}^{n} \tilde{K}_{ii}\, D_i Q D_i^*, \qquad V_R = \mathrm{tr}\sum_{i=1}^{n} \Delta K_{ii}\, D_i Q D_i^* \tag{5.5}$$

$$V_i = \mathit{Re}\ V_{ii}, \qquad V_{ii} = \mathrm{tr}\, K_{ii}\, D_i Q D_i^* \tag{5.6}$$

an orthornormal transformation can be found which satisfies the conditions of section 5.1. Hence, $T^*T = I$.

5.2 Cost Decoupled Components ($V_{ij} = V_{ii}\,\delta_{ij}$)

Observe from (3.3a) and (4.10) that the component costs associated with components x_i and x_j might have many common terms V_{ij}, $i \neq j$. If $V_{ij} = 0$ for all $i \neq j$ then the component costs (3.3a), (4.11) become "uncoupled" in the sense that (5.6) holds. Hence, the component costs V_i and V_j are independent. The class of transformations of interest in this section are those which yield property (5.6). The complete disturbance uncorrelation described by (5.4) provides one such transformation which yields property (5.6). Another choice is provided by the block diagonalization of K.

$$K = TK^oT^* = \text{block diag.}\{\dots K_{ii} \dots\},\ \ i=1,2,\ \dots\ n. \tag{5.7}$$

Thus, the orthonormal transformation T satisfying *either* (5.7) or (5.2) (with $D_R Q D_T^* = 0$) will yield uncoupled component costs of the form (5.6). The reader is reminded that the transformation induced by (5.7) is *different* from the one induced by (5.2)(with $D_R Q D_T^* = 0$) and will yield different values for (5.6). To summarize we cite

Corollary 2: Uncoupled Component Costs

If K *is block diagonal* (5.7) *then the* MEI *is given by* (4.7) *where*

$$\delta V = \sum_{i=1}^{n} \mathrm{tr}\, \tilde{K}_{ii}\, D_i Q D_i^*, \qquad V_R = \sum_{i=1}^{n} \mathrm{tr}\, \Delta K_{ii}\, D_i Q D_i^* \tag{5.8}$$

and V_i has the form (5.6).

5.3 Output Decoupled Components ($C_T = 0$)

In this section we consider those coordinates for which the output model error vector e_y in (1.2b) is zero. Any transformation of the original coordinates (5.1) which satisfies $C_T = 0$ has the property $e_y = 0$. This case is of interest in the examination of model degradation due only to state errors ($A_{RT} \neq 0$).

5.3 Output Decoupled Components ($C_T = 0$)

In this section we consider those coordinates for which the output model error vector C_y in (1.2b) is zero. Any transformation of the original coordinates (5.1) which satisfies $C_T = 0$ has the property $C_y = 0$. This case is of interest in the examination of model degradation due only to state errors ($A_{RT} \neq 0$).

Corollary 3: Model degradation due only to state errors ($A_{RT} \neq 0$).

If the output error is zero ($C_T = 0$) *then the* MEI *for the model* (4.8) *is given by* (4.7) *where*

$$\delta V = \mathrm{tr}[\tilde{K}_R D_R \mathcal{Q} D_R{}^* + \tilde{K}_{RT} D_T \mathcal{Q} D_R{}^* + \tilde{K}_{RT}^T D_R \mathcal{Q} D_T{}^* + \tilde{K}_T D_T \mathcal{Q} D_T{}^*] \qquad (5.9)$$

$$V_R = \mathrm{tr}[\Delta K_R D_R \mathcal{Q} D_R{}^* + \Delta K_{RT} D_T \mathcal{Q} D_R{}^* + \Delta K_{RT}^T D_R \mathcal{Q} D_T{}^* + \Delta K_T D_T \mathcal{Q} D_T{}^*] \qquad (5.10)$$

and the ΔK's *and* $\tilde{K}$'s *satisfy*

$$0 = K_{\sim} A_R + A_R{}^* K_{\sim} + C_R{}^* C_R \qquad (5.11)$$

$$0 = K A_R + A^* K + Q_3 \,, \quad Q_3 = \begin{bmatrix} 0 \\ A_{RT}^* K_{\sim} \end{bmatrix} , \quad K = \begin{bmatrix} K_1 \\ K_2 \end{bmatrix} \qquad (5.12)$$

$$0 = \tilde{K} A + A^* \tilde{K} + Q_4 \quad , \quad Q_4 = \left[\begin{array}{c|c} 0 & K_1 A_{RT} \\ \hline A_{RT}^* K_1{}^* & K_2 A_{RT} + A_{RT}^* K_2^* \end{array}\right] \qquad (5.13)$$

$$0 = \Delta K A + A^* \Delta K + Q_5 \,, \quad Q_5 = \left[\begin{array}{c|c} C_R^* C_R & -K_1 A_{RT} \\ \hline -A_{RT}^* K_1{}^* & -K_2 A_{RT} - A_{RT}^* K_2^* \end{array}\right] \qquad (5.14)$$

and the component cost is given by (4.10), *where* K *satisfies* (2.3) *with* $C_T=0$.

The output is more completely decoupled from the truncated components $x_T(t)$ if the output error e_y *and* the state error e_x are zero. The following corollary 4 holds only if x_T is unobservable in y.

Corollary 4: Truncation of unobservable components ($e_y = 0$, $e_x = 0$)

If the state and output errors are zero ($A_{RT} = 0$ *and* $C_T = 0$ in (5.1)) *then there is no model degradation,* MEI = 0.0, *and*

$$V_R = V = \mathrm{tr}\, \Delta K_R D_R \mathcal{Q}\, D_R^* \,, \quad \delta V = 0 \,, \; \tilde{K} = 0 \,, \qquad (5.15)$$

where ΔK_R *satisfies*

$$0 = \Delta K_R A_R + A_R^* \Delta K_R + C_R^* C_R \qquad (5.16a)$$

$$V_i = \mathrm{Re}\, V_{ii} \,, \quad V_{ij} = \mathrm{tr}[\Delta K_{R_{ij}} D_{R_j} \mathcal{Q} D_{R_i}^*] \qquad (5.16b)$$

There are many transformations T in (5.1) which satisfy $C_T = 0$. In an effort to choose that transformation that satisfies $C_T = 0$ *and* comes as near as possible to also satisfying $A_{RT} = 0$, ref. [3] chooses a generalized Hessenberg transformation that puts A_{RT} in a block triangular (Hessenberg) form and and selects $C_R = I$ in (5.1). This "output induced" transformation has the desirable property that $e_y = 0$ and, for the choice of coordinates satisfying $C_R = I$, x_R is as decoupled as possible from x_T. The model error index for such a generalized Hessenberg reduction (GHR) is given by corollary 3. Instead of the above choice of $A_{RT} \neq 0$, $C_T = 0$, one may also examine the coordinates which make $A_{RT} = 0$, $C_T \neq 0$ as described next.

5.4 Uncoupled Components

Now consider the case where the state error e_x is zero.

Corollary 5: Model degradation due only to output error ($e_y \neq 0$, $e_x = 0$)

If the state model error is zero ($A_{RT} = 0$) *then the* MEI *is given by* (4.7) *where*

$$\delta V = \mathrm{tr}[\tilde{K}_R D_R \mathcal{Q} D_R^* + \tilde{K}_{RT} D_T \mathcal{Q} D_R^* + \tilde{K}_{RT}^T D_R \mathcal{Q} D_T^* + \tilde{K}_T D_T \mathcal{Q} D_T^*] \tag{5.17a}$$

$$V_R = \mathrm{tr}[\Delta K_R D_R \mathcal{Q} D_R^* + \Delta K_{RT} D_T \mathcal{Q} D_R^* + \Delta K_{RT}^T D_R \mathcal{Q} D_T^*] \tag{5.17b}$$

and the ΔK's , $\tilde{K}$'s *satisfy*

$$0 = \tilde{K}_T A_T + A_T^* \tilde{K}_T + C_T^* C_T \tag{5.18a}$$

$$0 = \tilde{K}_{RT} A_T + A_R^* \tilde{K}_{RT} + A_{TR}^* \tilde{K}_T \tag{5.18b}$$

$$0 = \tilde{K}_R A_R + A_R^* \tilde{K}_R + (\tilde{K}_{RT} A_{TR} + A_{TR}^* \tilde{K}_{RT}^*) \tag{5.18c}$$

$$0 = \Delta K_{RT} A_T + A_R^* \Delta K_{RT} + C_R^* C_T \tag{5.18d}$$

$$0 = \Delta K_R A_R + A_R^* \Delta K_R + (C_R^* C_R + \Delta K_{RT} A_{TR} + A_{TR}^* \Delta K_{RT}^T) \tag{5.18e}$$

The component cost is given by (4.10) *where the solution of* (2.3) *for* K *simplifies to*

$$0 = K_T A_T + A_T^* K_T + C_T^* C_T \tag{5.19a}$$

$$0 = K_{RT} A_T + A_R^* K_{RT} + A_{TR}^* K_T + C_R^* C_T \tag{5.19b}$$

$$0 = K_R A_R + A_R^* K_R + K_{RT} A_{TR} + A_{TR}^* K_{RT}^T + C_R^* C_R \tag{5.19c}$$

a second circumstance in which $e_x = 0$ is described as follows.

Corollary 6: Components coupled only in the output ($A_{TR} = 0$, $A_{RT} = 0$)

If $A_{TR} = 0$ *and* $A_{RT} = 0$ *then the* MEI *is given by* (5.2) *where*

$$\delta V = \mathrm{tr}\, \tilde{K}_T D_T \mathcal{Q} D_T^* \tag{5.20a}$$

$$V_R = \mathrm{tr}[\Delta K_R D_R \mathcal{Q} D_R^* + \Delta K_{RT} D_T \mathcal{Q} D_R^* + \Delta K_{RT}^T D_R \mathcal{Q} D_T^*] \tag{5.20b}$$

and $\tilde{K}_T$ *and the* ΔK's *satisfy*

$$0 = \tilde{K}_T A_T + A_T^* \tilde{K}_T + C_T^* C_T \tag{5.21a}$$

$$0 = \Delta K_R A_R + A_R^* \Delta K_R + C_R^* C_R \tag{5.21b}$$

$$0 = \Delta K_{RT} A_T + A_R^* \Delta K_{RT} + C_R^* C_T \tag{5.21c}$$

The component cost is given by (4.10) *where* K *satisfies* (5.21) *where* $K_R = \Delta K_R$, $K_{RT} = \Delta K_{RT}$, $K_T = \tilde{K}_T$ *and*

$$K = \begin{bmatrix} K_R & K_{RT} \\ K_{RT}^* & K_T \end{bmatrix} \tag{5.22}$$

This section has shown the required computations for the MEI for a variety of coordinate choices for model truncation. A final choice of components for truncation, called "modal components", is a special case of corollary 6 and is of sufficient interest to warrant the space of the next section.

6.0 MODAL COST ANALYSIS (MCA)

This section is preoccupied with mathematical rather than physical component modeling, and the specific components to be examined are the "modal" components (which give A its Jordan form) obtained as the limiting case of corollary 6. The main result of this section is summarized as the following special case of theorem 4.1. First, we note that the component output matrices C_i (and the component disturbance matrices D_i) in the "real modal" coordinates (to be defined in what follows) will, respectively, have either one column (one row) if the ith eigenvalue is real, or two columns (two rows) if the ith eigenvalue is complex. To distinguish between the two types of C_i's and D_i's we will use the lower case letters to denote the vectors $\theta_i \triangleq C_i$ and $d_i{}^T = D_i$ only when the ith eigenvalue is real.

Corollary 7: Modal Components

Let A *have independent eigenvectors,* $A_{RT} = 0$, $A_{TR} = 0$, *and* A_R *and* A_T *be in their real Jordan forms*

$$A_R = \text{block diag.}\{\dots .\ J_i\ \dots .\} \qquad i=1,2,\dots\ r \tag{6.1a}$$

$$A_T = \text{block diag.}\{\dots .\ J_i\ \dots .\} \qquad i=r+1,\dots\ n \tag{6.1b}$$

where $J_i = \sigma_i$ *if the* ith *eigenvalue,* $\lambda_i = \sigma_i + j\omega_i$ *is real* $(\omega_i = 0)$ *and*

$$J_i = \begin{bmatrix} \sigma_i & \omega_i \\ -\omega_i & \sigma_i \end{bmatrix} \quad \text{if} \quad \omega_i > 0. \tag{6.1c}$$

Then the modal cost (cost associated with modal component x_i*) is given by*

$$V_i = \sum_{j=1}^{n} V_{ij} \tag{6.2}$$

where V_{ij} *has the following forms:*

- $\underline{\text{if } \omega_i = 0}$, $\underline{\omega_j = 0}$; *(denote* $\theta_i \triangleq C_i$, $d_i^T \triangleq D_i$*)*

$$V_{ij} = -(\sigma_i + \sigma_j)^{-1}\ \theta_i^T\theta_j\ d_j^T Q d_i \tag{6.3a}$$

- $\underline{\text{if } \omega_i = 0}$, $\underline{\omega_j > 0}$; *(define* $\sigma_{ij} \triangleq \sigma_i + \sigma_j$*)*

$$V_{ij} = -(\sigma_{ij}{}^2 + \omega_j{}^2)^{-1}\ \theta_i^T C_j[\sigma_i I + J_j^T]D_j Q d_i \tag{6.3b}$$

- $\underline{\text{if } \omega_i > 0}$, $\underline{\omega_i = 0}$;

$$V_{ij} = -(\sigma_{ij}{}^2 + \omega_i{}^2)^{-1}\ d_j^T Q D_i^T[\sigma_j I + J_i]C_i^T\theta_j \tag{6.3c}$$

- $\underline{\text{if } \omega_i > 0}$, $\underline{\omega_j > 0}$; *(denote* $C_i = [C_i{}^{(1)},\ C_i{}^{(2)}]$, $D_i^T = [d_i{}^{(1)},\ d_i{}^{(2)}]$

$$\begin{aligned} V_{ij} = {} & (\sigma_{ij}{}^2 + \omega_i{}^2)^{-1} C_j{}^{(1)^T} C_i[\sigma_j I + J_i^T]D_j Q d_i{}^{(1)} + \\ & [(\sigma_{ij}{}^2 + \omega_j{}^2 - \omega_i{}^2)^2 + 4\sigma_{ij}{}^2\omega_i{}^2]^{-1}\{(C_j{}^{(2)^T} C_i[\sigma_j I + J_i] - \omega_j C_j{}^{(1)^T} C_i) \cdot \\ & ([\sigma_j I + J_i^T]^2 + \omega_j^2 I)(D_j Q d_i{}^{(2)} + (\sigma_{ij}{}^2 + \omega_i^2)^{-1}\omega_j(\sigma_j I + J_i^T)D_j Q d_i^{(1)})\}. \end{aligned} \tag{6.3d}$$

The modal error index MEI *is given by* (4.7) *where*

$$\delta V = \sum_{i,j=r+1}^{n} V_{ij} \tag{6.4a}$$

$$V_R = \sum_{i=1}^{r}\left[\sum_{j=1}^{n} V_{ij}\right] + \sum_{i=r+1}^{n}\left[\sum_{j=1}^{r} V_{ij}\right] \tag{6.4b}$$

using the V_{ij} *from* (6.3).

The proof of corollary 7 follows in a straight-forward (although tedious) manner by obtaining the closed form solution of (2.3) when A is in the indicated real Jordan form (6.1). Although corollary 7 is specialized to the case of independent eigenvectors, the general theory is not. The case of generalized real Jordan forms (when generalized eigenvectors are present) requires too much space to write out V_{ij} explicitly in this paper, and the reader is referred to the general (although less detailed) formulas of theorem 4.1 for this case. For computer programming of the modal costs it is not necessary to have the computer perform the matrix manipulations in (6.3), since these can all be expanded by hand before programming (the matrices have no more than two rows or columns). Space does not warrant the detailed expansion of (6.3) here. Several special cases of corollary 7 are also of interest. These are shown in the next two corollaries.

Corollary 8: Uncoupled Modal Disturbances

If $D_j \mathcal{Q} D_i = 0$ *for* $i \neq j$ *then* (6.3) *reduces to:*

- *if* $\omega_i = 0$,

$$V_{ii} = -\tau_i \|\theta_i\|^2 \|d_i\|^2_{\mathcal{Q}} \;, \quad \tau_i = (2\,\sigma_i)^{-1} \tag{6.5a}$$

- *if* $\omega_i > 0$;

$$\begin{aligned} V_{ii} = {} & (\sigma_i{}^2 + \omega_i{}^2)^{-1} C_i{}^{(1)^T} C_i[\sigma_i I + J_i^T] D_i \mathcal{Q} d_i{}^{(1)} + \\ & \sigma_i{}^{-2}(\sigma_i{}^2 + 4\omega_i{}^2)^{-1}\{(C_i{}^{(2)^T} C_i[\sigma_i I + J_i] - \omega_i C_i^{(1)^T} C_i)\cdot \\ & [(\sigma_i I + J_i^T)^2 + \omega_i^2 I][D_i \mathcal{Q} d_i{}^{(2)} + (\sigma_i{}^2 + \omega_i{}^2)^{-1}\omega_i[\sigma_i I + J_i^T] D_i \mathcal{Q} d_i^{(1)}]\} \end{aligned} \tag{6.5b}$$

and

$$V_i = V_{ii} \;, \quad V = \sum_{i=1}^{n} V_i \tag{6.6}$$

$$\delta V = \sum_{i=r+1}^{n} V_i \;, \quad V_R = \sum_{i=1}^{r} V_i \;, \quad MEI = \delta V/(V_R + \delta V) \tag{6.7}$$

The MEI is minimized by truncating those modes with the smallest modal cost V_i.

For lightly damped systems such as flexible structures, σ_i and ω_j have the property $\sigma_i << \omega_j$ for all i,j. In this case the modal cost (6.2) reduces to

Corollary 9: Lightly Damped Systems

The modal cost $V_i \to V_{ii}$ *as* $\sigma_i/\omega_i \to 0$, *where*

$$V_i \equiv V_{ii} = \frac{1}{2}\tau_i\{\|C_i{}^{(1)}\|^2 + \|C_i{}^{(2)}\|^2\}\{\|d_i{}^{(1)}\|^2_{\mathcal{Q}} + \|d_i{}^{(2)}\|^2_{\mathcal{Q}}\} \tag{6.8}$$

and the MEI *is given by* (6.7) *using* (6.8). *The* MEI *is minimized by truncating modes with the smallest modal costs* V_i.

If the components are defined in the standard Jordan coordinates as in [4], instead of the real Jordan coordinates as in corollaries 7-9, then the modal costs V_i have the simpler expressions (compare (6.8) with eq. (50) of [4], using different notation) except that the observability vector θ_i and the disturbability vector d_i are *complex* in [4], whereas all vectors in (6.8) are real. Several observations concerning the structure of (6.5a) and (6.8) are worthy of note. In both cases V_i is related to three properties of each mode,

$$V_i = (\text{time constant})(\text{observability norm})(\text{disturbability norm}) \tag{6.9}$$

where mode i is observable in y if and only if its "observability norm" > 0 and mode i is disturbable from w if and only if its "disturbability norm" > 0. Thus the modal cost is zero for any mode which is either unobservable or undisturbable, or is arbitrarily far in the left half plane. Recalling that w in (2.2) includes both internal and external disturbances, suppose D' is zero in (2.1b), (2.2). Then all disturbances propagate through the actuators, in which case $BUB^* = D\mathcal{Q}D^*$ everywhere in this paper and B should replace D and U should replace $\mathcal{Q}$. Hence the modal cost (6.9) becomes the product of the *observability* and *controllability* norms of the mode. Since only the *product* is involved we need not concern ourselves with specialized normalizations which might be required to make these entities separately qualify as "measures" of controllability or observability in a strict mathematical sense. It is also of interest to note that the residue R_i of the mode i

$$C[s\ I - A]^{-1}D = \sum_{i=1}^{n} \frac{\theta_i d_i^*}{s-\lambda_i} \quad , \qquad R_i \triangleq \theta_i d_i^* \tag{6.10}$$

involves the *outer* product of its observability and disturbability (or controllability) vectors, while the modal costs (6.5a), (6.8) involve the *inner* products of these vectors. Of course, the residue R_i does not reflect the intensities of the various disturbances through $\mathcal{Q}$ as does the modal costs. Since the modal costs (6.8) provide a truncation criterion which minimizes the MEI for any specified order r, it follows that reduced order models obtained by matching residues possess MEI's equal to or greater than the MEI's obtained by modal cost truncation criteria.

7.0 CONCLUSIONS

A modal error index (MEI) is defined for the purpose of *evaluating* the quality of any given reduced model with respect to a second "evaluation model". To aid in the *construction* of reduced models the *component costs* V_i are defined to measure the participation of each component in the cost of the evaluation model. By choosing appropriate coordinate transformations prior to component truncation a number of different choices for "system components" are defined including those which cause decoupling of the outputs, decoupling of the disturbances, decoupling of the component dynamics, and decoupling of the component costs. Finally, when "modal" components are chosen for truncation according to the criterion of "modal costs" the model error index MEI is minimized. For lightly damped and certain other systems the modal costs reduce to a simple formula involving the product of observability and disturbability norm of each mode and the mode time constant. These truncation criteria are recommended for very large scale systems when only *open*-loop modal data can be generated for model reduction decisions. For lower order models the calculation of *closed*-loop modal data provide better truncation criteria [1].

REFERENCES

[1] Skelton, R. E., "Dynamic Controller Design Using Modal Cost Analysis", Proceedings of the CDC, Ft. Lauderdale, Florida, December 1979.

[2] Hirzinger, G. and G. Kreisselmeier, "On Optimal Approximation of High-Order Linear Systems by Low-Order Models", International Journal of Control, Vol. 22, No. 3, 1975, pp. 399-408.

[3] Edwin, C. Y., J. V. Medanic, and W. R. Perkins, "Generalized Hessenberg Transformations for Reduced-Order Modeling of Large-Scale Systems", Int. J. Control, Vol. 27, No. 4, pp. 493-512, 1978.

[4] Skelton, R. E. and C. Z. Gregory, "Measurement Feedback and Model Reduction by Modal Cost Analysis", Proceedings JACC, pp. 211-218, Denver, 1979.

SINGULAR PERTURBATIONS AND MODEL REDUCTION METHODS FOR LARGE SCALE AND DISTRIBUTED PARAMETER SYSTEMS

MARK J. BALAS
Electrical and Systems Engineering Department
Rensselaer Polytechnic Institute
Troy, New York 12181

ABSTRACT

Feedback control of large scale and distributed parameter systems must often be based on reduced-order models due to the very large or infinite dimension of the actual open-loop system. In this paper, we analyze the effects of controllers designed via reduced-order methods, including singular perturbations, and used to control the actual large scale or distributed parameter system. The results presented here provide bounds for stable operation; these bounds can be used to evaluate the modeling techniques and, if necessary, to redesign the controllers derived from them.

1.0 INTRODUCTION

Large scale systems are described by large dimensional systems of ordinary differential equations and distributed parameter systems by partial differential equations. Large scale systems include electric power distribution networks and econometric models; distributed parameter systems include heat diffusion and chemical processes, wave propagation, and mechanically flexible structures, such as large flexible spacecraft and satellites, high-speed aircraft and surface vehicles, and large civil engineering structures, e.g., bridges and tall buildings.

The state spaces for these systems have very large or infinite dimension in the case of large scale or distributed parameter systems, respectively. Thus, it is impractical or impossible to implement feedback controllers based on complete models of these systems; hence, reduced-order models must be used for the controller design. A great variety of reduced-order modeling techniques exist for general systems, e.g., the bibliography of [1], and new techniques are presently being developed in the specialized areas like large aerospace structures, e.g., [2], [3]. Of particular interest are the model reduction techniques based on asymptotic methods, such as multiple time scales and singular perturbations, e.g., [4]-[7] and the excellent survey [8]. These papers address the large scale but finite dimensional systems; to our knowledge, very little has been done with asymptotic methods for distributed parameter systems with the exceptions: [9] Chapt. 5, and the survey [10].

Our interest here is less in the area of derivation of reduced-order models (although that is an important topic in its own right); rather, we are concerned with the successful operation of the controller, designed on a reduced-order model, in closed-loop with the full system. Other work on this basic topic may be found, for example, in [2], [11]-[12], and, for singular perturbations methods, [13]-[17].

We consider linear, time-invariant large scale or distributed parameter systems with the following form:

$$\begin{cases} v_t = Av + Bf \quad , \quad v(0) = v_0 \\ y = Cv + Df \end{cases} \tag{1.1}$$

where the state v is in a Hilbert space H with inner product $(\cdot,\cdot)$ and associated norm $||\cdot||$ and the control vector f and observation vector y have dimensions M and P, respectively, which denote the number of (independent) actuators and sensors. When the system (1.1) is a distributed parameter system (DPS), the dim H = ∞ and the operator A is (usually) a differential operator with domain D(A), containing all states which satisfy the boundary conditions of the problem, dense in H while the input and output operators B and C have finite rank M and P, respectively, and D is a PxM matrix. The operator A generates a C_o semigroup U(t) on H. Correspondingly, when the system (1.1) is a large scale system, the dim H < ∞ but very large and the operators A, B, C, and D are all matrices with appropriate dimensions. Also, A generates a transition matrix U(t), i.e., $U(t) = e^{At}$, which is the finite dimensional analog of the semigroup in infinite dimensions. In either case, the semigroup U(t) is exponentially stable, i.e., it has the growth property:

$$||U(t)|| \leq M_o\, e^{-\sigma t} \quad , \quad t \geq 0 \tag{1.2}$$

where $M_o \geq 1$ and $\sigma > 0$. In most physical systems the usual energy dissipative mechanisms make (1.2) true, even for DPS; however, σ may be quite small, as it often is in the case of large aerospace structures [2].

In the following sections, we develop the general idea of reduced-order models for the systems of type (1.1) and obtain results on the operation of controllers, based on reduced-order models, in closed-loop with the actual system (1.1). For brevity, the proofs of these results will appear elsewhere.

2.0 REDUCED-ORDER MODELING AND CONTROLLER DESIGN

Let H_N and H_R be subspaces of the total state space H with dim $H_N = N < \infty$ and $H = H_N \oplus H_R$. Define the projection operators P_N and P_R (not necessarily orthogonal) and let $v_N = P_N v$ and $v_R = P_R v$. This decomposes v into $v = v_N + v_R$ and the system (1.1) into

$$\dot{v}_N = A_N v_N + A_{NR} v_R + B_N f \quad , \quad v_N(0) = P_N v_o \tag{2.1}$$

$$\dot{v}_R = A_{RN} v_N + A_R v_R + B_R f \quad , \quad v_R(0) = P_R v_o \tag{2.2}$$

$$y = C_N v_N + C_R v_R + Df \tag{2.3}$$

The reduced-order model (ROM) for the system is (2.1) and (2.3) with C_R and A_{NR} assumed to be zero:

$$\begin{cases} \dot{\tilde{v}}_N = A_N \tilde{v}_N + B_N f \quad , \quad \tilde{v}_N(0) = P_N v_o \\ \tilde{y} = C_N \tilde{v}_N + Df \end{cases} \tag{2.4}$$

Thus, the ROM depends on what choice of subspace H_N is made and what type of projection P_N is used (or, alternatively, what H_R is). The subspace H_N is called the ROM subspace and the subspace H_R, the residuals subspace. The terms $A_{NR} v_R$ and $A_{RN} v_N$ are called model error and $B_R f$ and $C_R v_R$ are called control and observation spillover, respectively.

The Controller Design is obtained by assuming the ROM (2.4) is the total system and using a linear controller of the form:

$$\begin{cases} f = H_{11}\, y + H_{12}\, z \\ \dot{z} = F\, z + Hy + Ef \quad , \quad z(0) = 0 \end{cases} \tag{2.5}$$

where dim $z = S \leq N$. Let $e \equiv z - T_N\, v_N$ and obtain (from (2.1), (2.3), and (2.5)):

$$\begin{cases} \dot{e} = F\, e + (H\, C_R - T_N\, A_{NR})\, v_R \\ e(0) = -T_N\, P_N\, v_o \end{cases} \tag{2.6}$$

when the following assumptions are made about T_N and E:

$$F\, T_N - T_N\, A_N + H\, C_N = 0 \tag{2.7}$$

$$E = T_N\, B_N - HD \tag{2.8}$$

There exists a unique solution T_N for (2.7), whenever F and A_N share no eigenvalues, [19] Chapt. 8. We will assume that F can have any desirable stability margin σ_F, i.e.,

$$||e^{Ft}|| \leq M_F\, e^{-\sigma_F t} \quad , \quad t \geq 0 \tag{2.9}$$

Using (2.3) in (2.5), we obtain

$$(I_M - H_{11}\, D)f = G_N\, v_N + H_{12}\, e + H_{11}\, C_R\, v_R \tag{2.10}$$

where

$$G_N = H_{11}\, C_N + H_{12}\, T_N \tag{2.11}$$

We will assume that the matrix $(I_M - H_{11}\, D)$ is nonsingular in order to explicitly obtain the control law

$$f = \tilde{G}_N\, v_N + \tilde{H}_{12}\, e + \tilde{H}_{11}\, C_R\, v_R \tag{2.12}$$

with $\tilde{G}_N = (I_M - H_{11}\, D)^{-1}\, G_N$, etc. In some cases we may need a further assumption to guarantee that H_{11}, H_{12} can be found to produce a predetermined set of gains G_N in (2.11); when this is necessary, we will make such an assumption.

Please note that all assumptions, e.g., (2.7)-(2.9), (2.11)-(2.12), have been made in terms of the parameters (A_N, B_N, C_N, D) of the ROM (2.4) and the controller parameters (H_{11}, H_{12}, F, H, E) of (2.5). No assumptions have been made yet on the residual subsystem parameters, the model error, or the spillover. Therefore, the controller design can be carried out entirely in terms of the ROM alone. However, the actual closed-loop system involves the residuals, too; thus, using (2.6) and (2.12) in

(2.1)-(2.2), we obtain:

$$\begin{cases} \dot{v}_N = (A_N + B_N \tilde{G}_N)\, v_N + B_N \tilde{H}_{12}\, e + (A_{NR} + B_N \tilde{H}_{11} C_R)\, v_R & (2.13) \\ \dot{e} = F\, e + (HC_R - T_N A_{NR})\, v_R & (2.14) \\ \dot{v}_R = (A_{RN} + B_R \tilde{G}_N)\, v_N + B_R \tilde{H}_{12}\, e + (A_R + B_R \tilde{H}_{11} C_R)\, v_R & (2.15) \end{cases}$$

with the initial conditions: $v_N(0) = P_N v_o$, $v_R(0) = P_R v_o$, and $e(0) = -T_N P_N v_o$.

The analysis of the closed-loop operation of (2.13)-(2.15) is carried out in the next section.

3.0 CLOSED-LOOP OPERATION OF THE ROM-BASED CONTROLLER

When a controller based on an ROM is designed as in Sec. 2.0, the closed-loop system with this controller is given by (2.13)-(2.15). In this section we assume the matrices $\tilde{A}_N \equiv A_N + B_N \tilde{G}_N$ and F have stability margins σ_N and σ_F, i.e.,

$$||e^{\tilde{A}_N t}|| \leq M_N e^{-\sigma_N t} \quad , \quad t \geq 0 \tag{3.1}$$

and

$$||e^{Ft}|| \leq M_F e^{-\sigma_F t} \quad , \quad t \geq 0 \tag{3.2}$$

and

$$\sigma_F > \sigma_N \tag{3.3}$$

Also, we assume the operator A_R generates a semigroup $U_R(t)$ on H_R with the growth property:

$$||U_R(t)|| \leq M_R e^{-\sigma_R t} \quad , \quad t \geq 0 \tag{3.4}$$

with $\sigma_R > 0$. Note that this semigroup $U_R(t)$ becomes the matrix exponential $e^{A_R t}$ when the residuals come from a large scale (i.e., dim H < ∞), as opposed to a distributed parameter system (dim H = ∞). All the constants M_N, M_F, and M_R are never less than unity.

We have the following useful result:

THEOREM 3.1: If no observation spillover or model error A_{NR} is present ($A_{NR} = 0$ and $C_R = 0$) in (2.13)-(2.15), then

$$\max(||v_N||, ||e||) \leq M_C e^{-\sigma_N t} ||P_N v_o|| \tag{3.5}$$

where

$$M_C \equiv M_N\, M_F\, \left(1 + \frac{||B_N\, \tilde{H}_{12}||}{\Delta_C}\right)$$

and

$$\Delta_C \equiv \sigma_F - \sigma_N \text{ (which is positive by (3.3)).}$$

In other words, the constant M_C can be determined directly from M_N and M_F. There are other constants M_C for which (3.5) holds, e.g., we could find the condition number of the Jordan form transformation of the matrix

$$\begin{bmatrix} \tilde{A}_N & B_N\, \tilde{H}_{12} \\ 0 & F \end{bmatrix}$$

In fact, the closer we can make M_C to unity in (3.5) (via whatever estimates) the better our later results become.

The following results give an indication of the closed-loop operation of a ROM-based controller with the total system:

THEOREM 3.2: Assume the model error terms A_{NR} and A_{RN} are zero. Assume $\sigma_N \neq \sigma_R$. When the observation spillover is zero, then the closed-loop system (2.13)-(2.15) has stability margin $\delta \equiv \min\,(\sigma_N, \sigma_R)$ and

$$||w|| \leq M_C\, M_R\, M_E\, e^{-\delta t}\, ||w_o|| \quad , \quad t \geq 0 \tag{3.6}$$

where

$$w \equiv [v_N{}^T\; e^T\; v_R{}^T]^T \quad , \quad v = v_N + v_R \text{, and}$$

$$||w|| \equiv (||v||^2 + ||e||^2)^{\frac{1}{2}} \quad \text{and } M_E \text{ is defined by}$$

$$M_E = \begin{cases} (1+\alpha)\; ||P_N|| + ||P_R|| & \text{if nonorthogonal projections } P_N \text{ and } P_R \text{ are used} \\ (1 + \alpha + \alpha^2)^{\frac{1}{2}} & \text{if orthogonal projections are used} \end{cases}$$

and $\alpha \equiv \beta/\Delta$, $\Delta \equiv |\sigma_N - \sigma_R|$, and the control spillover coefficient $\beta = ||B_R||\; (||\tilde{G}_N|| + ||\tilde{H}_{12}||)$.

THEOREM 3.3: Under the same assumptions as Theo. 3.2, except with observation spillover present, the closed-loop stability margin is at least δ' where

$$\delta' \equiv \delta - M_C\, M_R\, M_E\, \Gamma$$

when $\Gamma \equiv (||B||\ ||\tilde{H}_{11}\ C_R|| + ||H\ C_R||)$ is sufficiently small that $\delta' > 0$. In that case,

$$||w|| \leq M_C\ M_R\ M_E\ e^{-\delta' t}\ ||w_o|| \quad , \quad t \geq 0 \tag{3.7}$$

Note that in Theo. 3.2 and 3.3, since no model error is present, $\sigma_R = \sigma$ and $M_R = M_o$ where M_o and σ come from (1.2).

THEOREM 3.4: When the model error terms A_{NR} and A_{RN} are present, the result of Theo. 3.3 holds with $||A_{RN}||$ bounded and with Γ changed to $\Gamma + ||A_{NR}||\ (1 + ||T_N||)$ and β changed to $\beta + ||A_{RN}||$.

The proofs of these results will appear elsewhere. Note that the results of Theo. 3.1-3.3 appear in [20] for the special case of reducing subspaces. Other stability results for large scale and distributed parameter systems have appeared in [11] and [12]; connections between these and the above results are currently under investigation.

The value of the above results is that the bounds can be calculated from the ROM and from bounds on the model error and spillover, as long as the data in (3.4) is known. These results provide sufficient conditions for exponential stability of the closed-loop system even though the controller must be designed from some ROM of the actual system.

An alternative approach for certain ROMs obtained via singular perturbations is presented in the next section.

4.0 REDUCED-ORDER MODELING AND CONTROLLER DESIGN VIA SINGULAR PERTURBATIONS

For certain choices of subspaces H_N, H_R in Sec. 2.0, it is possible to produce ROM's such that (2.1)-(2.3) become:

$$\dot{v}_N = A_N^\varepsilon\ v_N + A_{NR}^\varepsilon\ v_R + B_N^\varepsilon f \tag{4.1}$$

$$\varepsilon\ \dot{v}_R = A_{RN}^\varepsilon\ v_N + A_R^\varepsilon\ v_R + B_R^\varepsilon f \tag{4.2}$$

$$y = C_N^\varepsilon\ v_N + C_R^\varepsilon\ v_R + Df \tag{4.3}$$

where the singular perturbation parameter $\varepsilon \geq 0$ represnets some small parameter dependence, such as electrical networks with parasitics or dynamical systems with small masses and time constants (see [21]). This approach is especially valuable when the ROM is based on vibration modes for a mechanically flexible structure and the frequency separation of slow and fast modes can be used to provide the decomposition of (4.1)-(4.3). An excellent survey of singular perturbation techniques for model reduction is presented in [8].

The parameters such as A_N^ε may involve regular perturbations in ε, as well, i.e.,

$$A_N^\varepsilon = A_N^0 + \varepsilon\ \tilde{A}_N^\varepsilon \tag{4.4}$$

where $\tilde{A}_N^{\varepsilon}$ is bounded for all ε and $\tilde{A}_N^{o} = 0$; however, D does not depend on ε in (4.3). Henceforth, we shall write A_N to denote A_N^{o}, etc. Although we do not consider this situation here, it is possible to have multi-parameter singular perturbations where several ε parameters are present; see, e.g., [22]-[23].

4.1 Large Scale Systems (dim H < ∞)

The ROM in this section is obtained from (4.1)-(4.3) by assuming that the fast system (4.2) has reached its steady-state with respect to the slow system (4.1), i.e., $\varepsilon \, \dot{v}_R \equiv 0$. Hence, take $\varepsilon = 0$ and obtain the singular perturbation ROM:

$$\begin{cases} \dot{\tilde{v}}_N = \bar{A}_N \, \tilde{v}_N + \bar{B}_N \, f \quad , \quad \tilde{v}_N(0) = P_N \, v_o \\ \tilde{y} = \bar{C}_N \, \tilde{v}_N + \bar{D} f \end{cases} \tag{4.5}$$

where $\bar{A}_N = A_N - A_{NR} \, A_R^{-1} \, A_{RN}$, $\bar{B}_N = B_N - A_{NR} \, A_R^{-1} \, B_R$, $\bar{C}_N = C_N - C_R \, A_R^{-1} \, A_{RN}$,

and $\bar{D} = D - C_R \, A_R^{-1} \, B_R$.

The above manipulation depends on the existence of A_R^{-1} and consequently we shall assume, henceforth, that A_R is nonsingular. This is not a serious limitation for most applications, and it can be weakened in some cases [24].

We note that the singular perturbations ROM (4.5) is different from the usual ROM (2.4) obtained in Sec. 2.0 in that correction terms are added to the original parameters (A_N, B_N, C_N, D) to obtain the ROM parameters ($\bar{A}_N$, $\bar{B}_N$, $\bar{C}_N$, $\bar{D}$). The only time these ROM's are the same is when the model error terms A_{NR} and A_{RN} are zero and either B_R or C_R is zero.

The controller design is based on the ROM (4.5) and is given by:

$$\begin{cases} f = H_{12} \, z \\ \dot{z} = \bar{A}_N \, z + \bar{B}_N \, f + H \, (y-\hat{y}) \quad , \quad z(0) = 0 \\ \hat{y} = \bar{C}_N \, z + \bar{D} \, f \end{cases} \tag{4.6}$$

This is a "full order" controller in the sense that it is the same order as that of the ROM (4.5), i.e., dim z = dim v_N; of course, it has a (much) lower order than that of the total system (4.1)-(4.3). The controller (4.6) may be compared to the one (2.5) obtained in Sec. 2.0 with $T_N = I_N$; in that case, (2.5) becomes:

$$\begin{cases} f = H_{12} \, z \\ \dot{z} = A_N \, z + B_N \, f + H \, (y-\hat{y}) \quad , \quad z(0) = 0 \\ \hat{y} = C_N \, z + D \, f \end{cases} \tag{4.7}$$

Although (4.6) and (4.7) have the same form, (4.6) may be rewritten as:

$$\begin{cases} f = H_{12}\, z \\ \dot{z} = A_N\, z + B_N\, F + H(y-\hat{y}) - A_{NR}\, q_R \\ \hat{y} = C_N\, z + D\, f - C_R\, q_R \\ q_R = A_R^{-1}\, (A_{RN}\, z + B_R\, f) \end{cases} \tag{4.8}$$

This clearly shows that the singular perturbations controller (4.6) is really the usual controller (4.7) with correction terms in q_R added (see (4.8)). Therefore, the controllers (4.6) and (4.7) are different, even in the special case of no model error ($A_{RN} = 0$ and $A_{NR} = 0$) when (4.8) becomes:

$$\begin{cases} f = H_{12}\, z \\ \dot{z} = A_N\, z + B_N\, f + H(y-\hat{y}) \\ \hat{y} = C_N\, z + D\, f - C_R\, q_R \\ q_R = A_R^{-1}\, B_R\, f \end{cases} \tag{4.9}$$

Note that (4.9) is equivalent to the following:

$$\begin{cases} f = H_{12}\, z \\ \dot{z} = (A_N + H\, C_R\, A_R^{-1}\, B_R\, H_{12})\, z + B_N\, f + H(y-\hat{y}) \\ \hat{y} = C_N\, z + D\, f \end{cases} \tag{4.10}$$

The controller (4.6) of this section is designed as though $\varepsilon = 0$, but it must operate "successfully" in closed-loop with the total system (4.1)-(4.3) when $\varepsilon > 0$ (although small). We have the following result which indicates that it does:

> THEOREM 4.1: Let $\dim H < \infty$. If the ROM $(\bar{A}_N, \bar{B}_N, \bar{C}_N)$ obtained in (4.5) is controllable and observable, then there exists a positive ε_0 such that controller gains H_{12} and H may be chosen so that, for every positive $\varepsilon \leq \varepsilon_0$, the total system (4.1)-(4.3) in closed-loop with the controller (4.6) based on the ROM will be asymptotically stable and the controller state z will converge to the system substate v_N as $t \to \infty$.

The proof of this theorem appears in [18] for the case where the parameters of (4.1)-(4.3) are constant; this proof can be easily extended to the case where the parameters vary as in (4.4). The proof is based on the Klimushchev-Krasovskii lemma ([25] or [26]) which is valid when $\dim H < \infty$ as we have assumed here; the validity of this lemma for $\dim H = \infty$ is still in question.

This result seems very useful for designing low-order controllers for certain large scale systems whose form is (4.1)-(4.3). It says that the controller will always operate successfully with the closed-loop system when ε is "small" enough. However, we must

(1) be able to formulate our large scale system in the format (4.1)-(4.3) so that the residual system involves the singular perturbation parameter ε and

(2) be able to say how small is "small."

These conditions mean that ε must be related to some physical parameter of the system so that we can make reasonable judgements about (1) and (2). For example, in [27] we have related ε to the separation of low frequency and high frequency spectra of mechanically flexible structures; this validates the idea that the controller can be designed based on the low frequency vibration modes alone, as long as sufficient spectral separation can be guaranteed.

4.2 Distributed Parameter Systems (dim H = ∞)

When dim H = ∞, the Klimushchev-Krasovskii lemma may not hold &, hence, the results of Theo. 4.1 are not available to us. However, in the special case where reducing subspaces are used for the ROM ([20]), the DPS of (4.1)-(4.3) has no model error ($A_{NR} = 0$ and $A_{RN} = 0$) and decomposes into the form:

$$\dot{v}_N = A_N v_N + B_N f \ , \ v_N(0) = P_N v_o \tag{4.11}$$

$$\varepsilon \dot{v}_R = A_R v_R + B_R f \ , \ v_R(0) = P_R v_o \tag{4.12}$$

$$y = C_N v_N + C_R v_R \tag{4.13}$$

where $\varepsilon \geq 0$ and the operator A_R generates a semigroup $U_R(t)$ with the same growth property (1.2) as that of U(t). We will assume, for simplicity, that the parameters of (4.11) - (4.13) are not functions of ε. The ROM is obtained when $\varepsilon = 0$ and is given by

$$\begin{cases} \dot{v}_N = A_N v_N + B_N f \ , \ v_N(0) = P_N v_o \\ y = C_N v_N + \overline{D} f \end{cases} \tag{4.14}$$

where $\overline{D} = -C_R A_R^{-1} B_R$. Note that A_R^{-1} exists and is bounded because of (1.2). The controller based on this ROM is finite dimensional and given by

$$\begin{cases} f = H_{12} z \\ \dot{z} = A_N z + B_N f + H(y-\hat{y}) \ , \ z(0) = 0 \\ \dot{y} = C_N z + \overline{D} f \end{cases} \tag{4.15}$$

This controller can be implemented for the DPS but its successful operation in closed-loop with the total system (4.11) - (4.13) when $\varepsilon > 0$ is governed by (using (4.15) in (4.11) - (4.13)):

$$\begin{bmatrix} \dot{h} \\ \varepsilon \dot{v}_R \end{bmatrix} = \begin{bmatrix} H_1 & H_2 \\ H_3 & H_4 \end{bmatrix} \begin{bmatrix} h \\ v_R \end{bmatrix} \tag{4.16}$$

where $h = [v_N^T \; e^T]^T$, $e = z - v_N$, and

$$\tilde{A}_N \equiv A_N - H\,C_N + H\,C_R A_R^{-1}\,B_R\,H_{12}\,,$$

$$H_1 = \begin{bmatrix} A_N + B_N H_{12} & B_N H_{12} \\ H\,C_R A_R^{-1}\,B_R H_{12} & \tilde{A}_N \end{bmatrix},$$

$$H_2 = \begin{bmatrix} 0 \\ HC_R \end{bmatrix}, \quad H_3 = [B_R H_{12} \quad B_R H_{12}] \; , \; H_4 = A_R \; .$$

Note that H_1, H_2, H_3 are all operators with finite rank and, consequently, they are bounded.

When $\varepsilon = 0$, (4.16) reduces to

$$\dot{h} = \bar{H}_1 h \tag{4.17}$$

where $\bar{H}_1 = H_1 - H_2\,H_4^{-1}\,H_3 = \begin{bmatrix} A_N + B_N H_{12} & B_N H_{12} \\ 0 & A_N - HC_N \end{bmatrix}$

which is stable by the choice of H and H_{12} when the ROM (A_N, B_N, C_N) is controllable and observable. The following result is the extension of Theo. 4.1 for DPS and it says that (4.16) remains stable for "small" values of $\varepsilon > 0$:

THEOREM 4.2: Let the gains H_{12} and H be chosen so that $\bar{H}_1$ in (4.17) has stability margin $\bar{\delta}$ and note that $H_4 = A_R$ generates a semigroup with the growth property (1.2). There exists $\varepsilon_0 > 0$ such that, for all $0 < \varepsilon \leq \varepsilon_0$, the closed-loop system, (4.11)-(4.13) and (4.15), is exponentially stable and the controller state z converges exponentially to the reduced state v_N.

The proof of this result will appear elsewhere; it uses extensions of results in [8] and [28] for matrices to the operators on a Hilbert space present here(via [29] Chapt. 3).

5.0 CONCLUSIONS

The results of Sec. 3.0 make it possible to design low-order controllers for large scale and distributed parameter systems via very general reduced-order models and analyze their operation in closed-loop with the actual system. The stability conditions presented in Theos. 3.1 - 3.4 can be checked with only knowledge of certain bounds on the unmodeled residuals of the open-loop system.

The results of Sec. 4.0 are for those special reduced-order models obtained via singular perturbations. For large scale systems the stability results are already known (Theo. 4.1) but, for distributed parameter systems, the result Theo. 4.2 is new and indicates that stability for the reduced-order system ($\varepsilon = 0$) remains valid for small values of the perturbation parameter ε. However, we hasten to point out that, although these results allay some of our fear in using a design for $\varepsilon = 0$ to control a system where ε is small but non-zero, they should not be used to put our guard completely to sleep; the basic problem remains: what is "small"? To answer this question requires that the perturbation parameter ε be related to physical quantities in the particular application so that "smallness" can be clearly understood. What is precisely small ε for the theorems to hold may not be related to obviously small physical quantities in a given application.

6.0 REFERENCES

1. R. Genesio and M. Milanese, A Note on the Derivation and Use of Reduced-Order Models, IEEE Trans. Autom. Contr., AC-21 (1976), 118-122.
2. M. Balas, Some Trends in Large Space Structure Control Theory: Fondest Hopes; Wildest Dreams, Invited Survey for Joint Autom. Contr. Conf., Denver, CO, 1979.
3. R. Skelton, Modal Cost Analysis in Model and Controller Reduction, 17th Allerton Conf. on Comm., Contr., and Computing, Monticello, IL, 1979.
4. P. Kokotović and W. Perkins, eds., Singular Perturbations-Order Reduction in Control System Design, A Symp. of the Am. Autom. Contr. Council, 1972 Joint Autom. Contr. Conf., Stanford, CA, 1972.
5. P. Kokotović and P. Sannuti, Singular Perturbation Method for Reducing the Model Order in Optimal Control Design, IEEE Trans. Autom. Contr., AC-13 (1968), 377-384.
6. R. Ramnath and G. Sundri, A Generalized Multiple Time Scales Approach to a Class of Linear Differential Equations, J. Math. Analy. and Appl., 28 (1969).
7. D. Teneketzis and N. Sandell, Linear Regulator Design for Stochastic Systems by a Multiple Time-Scales Method, IEEE Trans. Autom. Contr., AC-22 (1977), 615-621.
8. P. Kokotović, R. O'Malley, and P. Sannuti, Singular Perturbations and Order Reduction in Control Theory-An Overview, Automatica, 12 (1976), 123-132.
9. J. L. Lions, Some Aspects of the Optimal Control of Distributed Parameter Systems, SIAM, Philadelphia, PA, 1972.
10. J. L. Lions, Asymptotic Methods in the Optimal Control of Distributed Systems, Automatica, 14(1978), 199-211.
11. D. D. Siljak, Large Scale Dynamic Systems: Stability and Structure, Elsevier-North Holland Inc., N.Y., 1977.
12. A. N. Michel and R. K. Miller, Qualitative Analysis of Large Scale Dynamical Systems, Academic Press, N.Y., 1977.
13. M. Suzuki and M. Miura, Stabilizing Feedback Controllers for Singularly Perturbed Linear Constant Systems, IEEE Trans. Autom. Contr., AC-21 (1976), 123-124.
14. K. D. Young, Feedback Design of Singularly Perturbed Systems, 17th Allerton Conf. on Comm., Contr., and Computing, Monticello, IL, 1979.
15. S. Campbell, On a Singularly Perturbed Autonomous Linear Control Problem, IEEE Trans. Autom. Contr., AC-24(1979), 115-117.
16. N. Sandell, Robust Stability of Systems with Application to Singular Perturbations, Automatica, 15(1979), 467-470.
17. J. Sesak, P. Likins, and T. Coradetti, Flexible Spacecraft Control by Model Error Sensitivity Suppression, The J. of the Astronautical Sciences, 27(1979), 131-156.
18. M. Balas, Observer Stabilization of Singularly Perturbed Systems, AIAA J. Guidance and Control, 1(1978), 93-95.

19. F. Gantmacher, The Theory of Matrices, Vol. I. Chelsea, N.Y., 1959.
20. M. Balas, Feedback Control of Dissipative Distributed Parameter Systems, IEEE Conf. on Control and Decision, Ft. Lauderdale, FL, 1979.
21. P. Kokotović, A Control Engineer's Introduction to Singular Perturbations, contained in Ref. 4.
22. H. Khalil and P. Kokotović, Control of Linear Systems with Multiparameter Singular Perturbations, Automatica, 15(1979), 197-207.
23. H. Khalil and P. Kokotović, D-Stability and Multiparameter Singular Perturbation, SIAM J. Contr. and Opt., 17(1979), 56-65.
24. R. O'Malley and J. Flaherty, Singular Singular-Perturbation Problems, in Lecture Notes in Mathematics, Vol. 594, Springer-Verlag, N.Y., 1977.
25. A. Klimushchev and N. Krasovskii, Uniform Asymptotic Stability of Systems of Differential Equations with Small Parameter in the Derivative Terms, J. Appl. Math. and Mech., 25(1961), 1011-1025.
26. M. Shensa, Parasitics and the Stability of Equilibrium Points of Nonlinear Networks, IEEE Trans. Cir. Theory, CT-18(1971), 481-484.
27. M. Balas, Lecture Notes on Control of Large Mechanically Flexible Structures, (unpublished manuscript).
28. P. Kokotović, A Riccati Equation for Block-Diagonalization of Ill-Conditioned Systems, IEEE Trans. Autom. Contr., AC-20(1975), 812-814.
29. A. Pazy, Semigroups of Linear Operators and Applications to Partial Differential Equations, University of Maryland, Lecture Note #10, 1974.

Appendix: Proof of Theo. 4.2

In (4.16), make the change of variables (e.g. [8]):

$$\xi_R = v_R + Q_3 h \tag{4.18}$$

where $Q_3 = H_4^{-1} H_3 + \varepsilon L_1$ and obtain:

$$\dot{h} = (\overline{H}_1 + \varepsilon Q_1)h + H_2 \xi_R \tag{4.19}$$

$$\varepsilon \dot{\xi}_R = (H_4 + \varepsilon Q_z)\xi_R \tag{4.20}$$

where $Q_1 = -H_2 L_1$, $Q_2 = H_4^{-1} H_3 + \varepsilon L_1$, and L_1 satisfies:

$$H_4^{-1} H_3 \overline{H}_1 + \varepsilon L_1 \overline{H}_1 - (H_4 + \varepsilon H_4^{-1} H_3 H_2)L_1 - \varepsilon^2 L_1 H_2 L_1 = 0 \tag{4.21}$$

Let ε_0 be sufficiently small that

$$\varepsilon_0 \, ||H_4^{-1}|| \leq \frac{1}{3} (||\overline{H}_1|| + ||H_2|| \, ||H_4^{-1} H_3||)^{-1} \tag{4.22}$$

then, for all $0 < \varepsilon \leq \varepsilon_0$, a unique bounded solution L_1 exists for (4.21) by taking $A_{11} = H_1$, $A_{12} = H_2$, $\varepsilon A_{21} = H_3$, $\varepsilon A_{22} = H_4$, $D = \varepsilon L_1$, $A_o = \overline{H}_1$, and $L_o = H_4^{-1} H_3$ in Lemma 1 of [28]; this lemma also guarantees that the bound on L_1 is given by:

$$||L_1|| \leq \frac{2}{\varepsilon} \; \frac{||\overline{H}_1|| + ||H_4^{-1} H_3||}{||\overline{H}_1|| + ||H_2|| \, ||H_4^{-1} H_3||} \tag{4.23}$$

Note that the proof of Lemma 1 in [28] is done by a contraction mapping argument and, hence, it remains valid for our function space needs here, even though it is only stated in terms of matrices in [28].

Now change the time scale in (4.20) to $\tau = t/\varepsilon$ and define $\tilde{\xi}_R(\tau) \equiv \xi_R(t/\varepsilon)$. Therefore, from (4.20), obtain:

$$\frac{d\tilde{\xi}_R(\tau)}{d\tau} = \varepsilon \frac{d\xi_R(t/\varepsilon)}{dt} = (H_4 + \varepsilon Q_2) \, \tilde{\xi}_R(\tau) \tag{4.24}$$

with $\tilde{\xi}_R(0) = \xi_R(0)$. Since H_4 generates an exponentially stable semigroup by hypothesis and Q_2 is bounded, from [29] Chapt. 3, $H_4 + \varepsilon Q_2$ generates a semigroup $U_4(t)$ with the growth property:

$$||U_4(t)|| \leq M_4 \, e^{-\delta_4 t} \quad , \; t \geq 0 \tag{4.25}$$

when ε is sufficiently small that

$$\delta - \varepsilon M_4 \, ||Q_2|| > 0 \tag{4.26}$$

where δ comes from (1.2). Although $\overline{H}_1 + \varepsilon Q_1$ has finite rank, a similar argument shows that it generates a semigroup (transition matrix) $U_1(t)$ with the property that:

$$||U_1(t)|| \leq M_1 e^{-\delta_1 t}, \quad t \geq 0 \tag{4.27}$$

when ε is sufficiently small that

$$\overline{\delta} - \varepsilon M_1 ||Q_1|| > 0 \tag{4.28}$$

Let ε_0 be chosen small enough that (4.26) and (4.28) are satisfied.

From (4.24) and (4.25), obtain

$$||\xi_R(t/\varepsilon)|| \leq M_4 e^{-\frac{\delta_4 t}{\varepsilon}} ||\xi_R(0)|| \tag{4.29}$$

or, by rescaling t/ε to t,

$$||\xi_R(t)|| \leq M_4 e^{-\delta_4 t} ||\xi_R(0)|| \tag{4.30}$$

Also, for $t \geq 0$,

$$h(t) = U_1(t)\, h(0) + \int_0^t U_1(t-t')\, H_2\, \xi_R(t')dt' \tag{4.31}$$

and, using (4.25), (4.27), and (4.30),

$$||h(t)|| \leq M_1 e^{-\delta_1 t} [||h(0)|| + M_4 ||H_2|| \, ||\xi_R(0)|| \int_0^t e^{(\delta_1 - \delta_4)t'} dt' \tag{4.32}$$

Let $\Delta \equiv |\delta_1 - \delta_4|$, If $\delta_1 < \delta_4$, (4.32) yields

$$||h(t)|| \leq M_1 e^{-\delta_1 t} [||h(0)|| + M_4 ||H_2|| \, ||\xi_R(0)|| \frac{1-e^{-\Delta t}}{\Delta} \tag{4.33}$$

However, if $\delta_4 < \delta_1$, (4.32) becomes

$$\begin{aligned} ||h(t)|| &\leq M_1 e^{-(\delta_4 + \Delta)t} [||h(0)|| + M_4 ||H_2|| \, ||\xi_R(0)|| \frac{e^{\Delta t}-1}{\Delta}] \\ &\leq M_1 e^{-\delta_4 t} [||h(0)|| + M_4 ||H_2|| \, ||\xi_R(0)|| \frac{1-e^{-\Delta t}}{\Delta} \end{aligned} \tag{4.34}$$

From either (4.33) or (4.34) obtain:

$$||h(t)|| \leq \tilde{M} e^{-\tilde{\delta} t} [||h(0)|| + ||\xi_R(0)||] \tag{4.35}$$

where $\tilde{\delta} \equiv \min(\delta_1, \delta_4)$ and $\tilde{M} \equiv (M_1 M_4) \max(1, \frac{||H_2||}{\Delta})$.

Therefore, from (4.30) and (4.35) obtain:

$$||h(t)|| + ||\xi_R(t)|| \leq 2\tilde{M} e^{-\tilde{\delta} t} [||h(0)|| + ||\xi_R(0)||] \tag{4.36}$$

However, from (4.18) obtain:

$$||v_R|| \le ||\xi_R|| + ||Q_3||\ ||h|| \qquad (4.37)$$

and

$$||\xi_R(0)|| \le ||v_R(0)|| + ||Q_3||\ ||h(0)|| \qquad (4.38)$$

and, therefore (using (4.37) - (4.38) in (4.36)):

$$||h(t)|| + ||v_R(t)|| \le 2\ (1 + ||Q_3||)^2\ \tilde{M}\ e^{-\tilde{\delta}t}[||h(0)||+||v_R(0)||] \qquad (4.39)$$

This guarantees the exponential stability of (4.16) for $0 < \varepsilon \le \varepsilon_0$. ■

FEEDBACK DESIGN OF SINGULARLY PERTURBED SYSTEMS[*]

K-K. DAVID YOUNG
Dept. of Mechanical Engr. and Mechanics
Drexel University
Philadelphia, Pa. 19104

ABSTRACT

The robustness of asymptotic root-loci and cheap optimal quadratic regulator feedback design methods with respect to neglected fast dynamics is examined.

INTRODUCTION

Singular Perturbation methods have been successfully applied to decompose the design of the full order system into feedback designs of a slow and a fast subsystem [1, 2]. The slow subsystem and its associated feedback design can be separated from the fast subsystem. Design of feedback systems often involves only the slow subsystem design. The slow subsystem serves as a reduced order design model which is to approximate the full order system. The spirit of the separation of design lies in maintaining the magnitude of each open loop eigenvalue while changing its value through feedback. Thus it is prohibited in slow subsystem design to shift any of the slow open loop eigenvalue to be as large as any of the fast open loop eigenvalue. In contrast, recently developed feedback design methods which include cheap optimal quadratic regulator [3, 4] and asymptotic root loci method [5] deploy large feedback gains which forces the open loop eigenvalues to form groups of various magnitudes. If reduced order design model is used, these design methods force some of the reduced order model eigenvalues to be as large as the fast eigenvalues which are neglected in the model. Thus the applicability of these design methods to practical feedback design problem is determined by its robustness with respect to the neglected fast dynamics. This paper addresses some of the issues that arise when attempt is made to answer the following robustness question: if a reduced order design model is used in an asymptotic root loci (or a cheap optimal quadratic regulator) feedback design how would the designed low frequency behavior characterized by the transmission zeros and the high frequency behavior determined by the root

---

[*] This work is in part supported by U.S. Department of Energy under Contract no. ET-78-S-01-3089.

loci asymptotes differ from that of the full order feedback system?

We begin with specifying the full order open loop model,

$$\dot{x}_1 = A_{11}x_1 + A_{12}x_2 + B_1u \tag{1}$$

$$\mu\dot{x}_2 = A_{21}x_1 + A_{22}x_2 + B_2u \tag{2}$$

with the assumption that A_{22} is a Huwitz matrix. We define the reduced order design model for (1), (2) to be

$$\dot{\bar{x}}_1 = \bar{A}\bar{x}_1 + \bar{B}u \tag{3}$$

$$\bar{A} \equiv A_{11} - A_{12}A_{22}^{-1}A_{21} \tag{4}$$

$$\bar{B} \equiv B_1 - A_{12}A_{22}^{-1}B_2 \tag{5}$$

System (3) is the slow subsystem of the full order system. An asymptotic root lori feedback design which is based on (3) is denoted by

$$u = g\bar{K}\bar{x}_1 \tag{6}$$

where g is a large positive scalar. With $\bar{x}_1$ replaced by x_1 the feedback control applied to the full order system is obtained,

$$u = g\bar{K}x_1 \tag{7}$$

The examination of the robustness issue involves comparison of the eigenvalues of the reduced system (3), (6) and that of the full order system (1), (2), (7). In order to focus on the robustness issue we assume that $\bar{K}\bar{B}$ is nonsingular in the following discussions. From [6], this assumption implies that the designed eigenvalues of (3), (6) consists of only two groups with $0(1)$ and $0(g)$ magnitudes.

TRANSMISSION ZEROS OF FULL ORDER SYSTEM

For the reduced system, the transmission zeros of $(\bar{K}, \bar{A}, \bar{B})$ which are the finite closed loop eigenvalues as g tends to infinity have been shown [6] to be $\lambda(\bar{M}\bar{A}\bar{N})$ where

$$\bar{M}\bar{B} = 0 \tag{8}$$

$$\bar{K}\bar{N} = 0 \tag{9}$$

For the full order system, the transmission zeros of the triple

$$\left([\bar{K} \quad 0], \begin{bmatrix} A_{11} & A_{12} \\ \dfrac{A_{21}}{\mu} & \dfrac{A_{22}}{\mu} \end{bmatrix}, \begin{bmatrix} B_1 \\ \dfrac{B_2}{\mu} \end{bmatrix} \right) \tag{10}$$

are by definition those λ such that

$$\text{rank}\begin{bmatrix} A_{11} - \lambda I & A_{12} & B_1 \\ \frac{A_{21}}{\mu} & \frac{A_{22}}{\mu} - \lambda I & \frac{B_2}{\mu} \\ \bar{K} & 0 & 0 \end{bmatrix} < n_1 + n_2 + m \qquad (11)$$

In general it is difficult to use (11) directly in determining the transmission zeros. Since the full order system is a singularly perturbed system, it has been shown [2, 7] that there exists a state transformation which transforms (x_1, x_2) into

$$\dot{\xi} = (\bar{A} - \mu A_{12}Q)\xi + (\bar{B} + \mu P)u \qquad (12)$$

$$\dot{\eta} = \left(\frac{A_{22}}{\mu} + LA_{12}\right)\eta + \left(\frac{B_2}{\mu} + LB_1\right)u \qquad (13)$$

where ξ and η are the so-called slow and fast variables and L, P and Q are properly defined matrices. If an output y for the full order system is defined,

$$y = [\bar{K} \quad 0]\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad (14)$$

In terms of ξ and η

$$y = [\bar{K} \quad \mu\bar{K}H]\begin{bmatrix} \xi \\ \eta \end{bmatrix} \qquad (15)$$

The transmission zeros of the full order system with respect to the input-output pair (u, y) are invariant with respect to state transformation. The transformed system (12), (13), (15) can be used to determine the transmission zeros. Since the ξ and η dynamics are decoupled, as $\mu \to 0$ the fast variable η is unobservable through y. Using the results in [8], $\lambda\left(\frac{A_{22}}{\mu}\right)$ and the transmission zeros of $(\bar{K}, \bar{A}, \bar{B})$ are the transmission zeros of the full order system. Thus if a set of finite eigenvalues, λ_{fin} is designed using the reduced order model, this set remains to be the finite eigenvalues of the full order feedback system in the limit as $\mu \to 0$. We therefore conclude that the design of λ_{fin} is robust.

It is of interest also to approximate the finite eigenvalues of the full order system for small μ. We note that if $A_{12} = 0$, the transmission zeros of (10) for any μ are precisely those obtained by tending μ to zero. To find an approximation to λ_{fin} of the full order system, we use the fact that λ_{fin} coincides with the finite eigenvalues of the closed loop feedback system as $g \to \infty$. The closed loop system matrix is

$$F(g, \mu) = \begin{bmatrix} A_{11} + gB_1\bar{K} & A_{12} \\ \frac{A_{21}}{\mu} + \frac{g}{\mu} B_2\bar{K} & \frac{A_{22}}{\mu} \end{bmatrix} \quad (16)$$

In this analysis of the limiting eigenvalues of F as $g \to \infty$ we need to impose the condition

$$\lim_{\substack{g \to \infty \\ \mu \to 0}} \frac{1}{g\mu} = 0 \quad (17)$$

which means g tends to infinity faster than μ tends to zero. This condition assures that λ_{fin} are the transmission zeros of the full order system. Taking a singular perturbation approach, we found that a singular singular perturbation problem has to be solved for the general case. Singular singular perturbation problems have been considered in [9, 10, 11] however their solutions are not applicable. For two special cases, we have obtained power series expansion of λ_{fin} in μ, they are when (i) $B_2 = 0$, (ii) $B_1 = 0$ and B_2 is nonsingular. The first case reduces to a regualr singular perturbation problem. The second case results in a particular singular singular perturbation problem which has been solved earlier [12]. The system structure of case (ii) is often encountered when the fast subsystem represents parasitic actuator dynamics.

LARGE EIGENVALUES OF FULL ORDER SYSTEM

The high frequency behavior of the full order feedback system is determined by its large eigenvalues which can be calculated from the approximations of the infinite eigenvalues λ_{inf} of $F(g, \mu)$ as $g \to \infty$. For the same reason given earlier, only results for the two special cases are obtained. Nevertheless from these results the robustness of the high frequency behavior is deduced.

For case (i), a singular perturbation analysis of (16) shows that the infinite eigenvalues of the full order feedback system can be approximated by $g\lambda(\bar{K}\bar{B})$ and $\lambda(\frac{A_{22}}{\mu})$. From [6], under our assumption that $\bar{K}\bar{B}$ is nonsingular, the infinite eigenvalues designed using the reduced order model is $g\lambda(\bar{K}\bar{B})$. Thus in this case the designed infinite eigenvalues remain to be the λ_{inf} of the full order feedback system. The conclusion is that for case (i) the asymptotic root loci feedback design based on the reduced order model is robust.

In case (ii), if the nonsingular matrix $\bar{K}A_{12}A_{22}^{-1}B_2$ is diagonalizable, using a singular perturbation technique given in [12], it can be shown that λ_{inf} of the full order feedback system are approximated by

$$\lambda_{inf}^{i} = \frac{r_i}{\mu} \pm j\sqrt{\frac{g}{\mu}}\sqrt{w_i^2 - \frac{\mu}{g}r_i^2}\ , \quad i = 1, \ldots, m \tag{18}$$

where

$$\Gamma\bar{K}A_{12}A_{22}^{-1}B_2\Gamma^{-1} = \mathrm{diag}\ (w_1^2, \ldots, w_m^2) \tag{19}$$

$$\Gamma A_{22}\Gamma^{-1} = \mathrm{diag}\ (2r_1, \ldots, 2r_m) \tag{20}$$

Since the designed infinite eigenvalues using a reduced order model are $g\lambda(\bar{K}\bar{B})$, (18) shows that the large eigenvalues of the full order feedback system differ from their designed values significantly. This result shows that an asymptotoic root loci feedback design based on a reduced order model need not be robust with respect to the neglected fast dynamics. We note that r_i in (18) need not be negative. Thus instability may arise from this reduced order design.

CONCLUSIONS

The robustness of asymptotic root loci/cheap optimal regulator feedback design method with respect to neglected fast dynamics is examined.

We found that the designed finite eigenvalues in general remain to be the finite eigenvalues of the full order feedback system, however the characteristics of the infinite eigenvalues may be totally different from that of the designed infinite eigenvalues as shown in one of the cases considered. The results shown in this paper suggest that extreme care must be exercised in choosing a reduced order model for these types of feedback design.

REFERENCES

[1] P.V. Kokotovic, R.E. O'Malley, Jr., and P. Sannuti, "Singular Perturbations and Order Reduction in Control Theory - an Overview," Automatica, Vol. 12, 1976, pp. 123-132.

[2] J.H. Chow and P.V. Kokotovic, "A Decomposition of Near-Optimum Regulators for Systems with Slow and Fast Modes," IEEE Trans. on Automatic Control, Vol. AC-21, 1976, pp. 701-705.

[3] C.H. Harvey and G.Stein, "Quadratic Weights for Asymptotic Regulator Properties," IEEE Trans. on Automatic Control, Vol. AC-23, pp. 378-387, 1978.

[4] G. Stein, "Generalized Quadratic Weights for Asymptotic Regulator Properties," IEEE Trans. on Automatic Control, Vol. AC-24, August 1979.

[5] B. Kouvaritakis, "Gain Margins and Root-Locus Asymptotic Behavior in Multivariable Design," International Journal Control, Vol. 27, No. 5, pp. 705-724, 1978.

[6] K-K. D. Young, P.V. Kokotovic and V.I. Utkin, "A Singular Perturbation Analysis of High Gain Feedback Systems," IEEE Trans. on Automatic Control, Vol. AC-22, pp. 931-938, 1977.

[7] P.V. Kokotovic, "A Riccati Equation for Block Diagonalization of Ill-Conditioned System," IEEE Trans. on Automatic Control, Vol. AC-20, pp. 812-814, 1975.

[8] E.J. Davison and S.H. Wang, "Properties and Calculation of Transmission Zeros of Linear Multivariable Systems," Automatica, Vol. 10, pp. 643-658, 1974.

[9] A.B. Vasileva, "Singularly Perturbed Systems with an Indeterminacy in Their Degenerate Equations," Differential Equations, Vol. 12, pp. 1227-1235, 1976.

[10] R.E. O'Malley, Jr., and J.E. Flaherty, "Singular Singular-Perturbations and Boundary Layer Theory, Lyon 1976, pp. 422-436, Springer Verlag, New York 1977.

[11] S.L. Campbell, On a Singularly Perturbed Autonomous Linear Control Problem," IEEE Trans. on Automatic Control, Vol. AC-24, pp. 115-117, 1979.

[12] K-K. D. Young and P.V. Kokotovic, "Actuator and Sensor Dynamics in Multivariable Feedback Systems," Proc. 1978 IEEE Conf. on Decision and Control, pp. 868-870.

A FUNCTIONAL ANALYTIC APPROACH TO AUTONOMOUS LINEAR SINGULAR SINGULAR PERTURBATION PROBLEMS

STEPHEN L. CAMPBELL

Department of Mathematics, North Carolina State University, Raleigh, NC

ABSTRACT

Traditionally the relationship between a full system $A(\varepsilon)\dot{\underline{x}} + B(\varepsilon)\underline{x} = \underline{f}(\varepsilon,t)$ and its reduced form, $A(0)\dot{\underline{x}} + B(0)\underline{x} = \underline{f}(0,t)$ is investigated by the use of matched asymptotic expansions. This paper will explain and discuss an alternative approach. The strengths and weaknesses of this approach will be presented as well as sample results.

INTRODUCTION

Let $\underline{x}(t)$, $\underline{f}(\varepsilon,t)$ be n-vectors. Let $A(\varepsilon)$, $B(\varepsilon)$ be n×n matrices. Here t,ε are real variables. For convenience we assume $\underline{f}$, A,B are analytic in ε. The full system of differential equations

$$A(\varepsilon)\dot{\underline{x}} + B(\varepsilon)\underline{x} = \underline{f}(\varepsilon,t) \tag{1}$$

appears in a variety of control problems. The reduced problem

$$A(0)\dot{\underline{x}} + B(0)\underline{x} = \underline{f}(0,t) \tag{2}$$

is singular if A(0) is not invertible. The mathematical problem is to relate solutions of the reduced problem to those of the full problem. There are several practical reasons for doing this. One is to examine the effect of small parameters and investigate boundary layers. A second is to try and build solutions for the full equations out of solutions to the lower order reduced problem.

Traditionally these problems are usually solved by constructing series in slow variables for the interior of the region of interest and series in fast variables for the boundary regions and then matching up these series to get a solution. Problems of convergence are not always simply dealt with. This approach has been widely used, see for example, [2], [14], [17]-[27] and has proved quite useful. Results of this type usually involve sufficient, but not necessary conditions.

For linear problems of the form (1), (2) we have developed an alternative approach which differs in several ways from the traditional approach. The technical details of this approach may be found in [4], [5], [10], [11], and [6]. This paper will instead discuss the intuitive motivation for these techniques since they involve mathematical machinery from functional analysis. The strengths and shortcomings of this approach will be discussed.

FUNCTIONAL ANALYTIC BACKGROUND

If $C(\varepsilon)$ is a matrix valued function and the eigenvalues of $C(\varepsilon)$, denoted $\sigma(C(\varepsilon))$, fall into two disjoint sets $\delta_1(\varepsilon)$, $\delta_2(\varepsilon)$, then there is an invertible matrix $T(\varepsilon)$ so that

$$C(\varepsilon) = T(\varepsilon)\begin{bmatrix} C_1(\varepsilon) & 0 \\ 0 & C_2(\varepsilon) \end{bmatrix} T(\varepsilon)^{-1} \tag{3}$$

where $\sigma(C_i(\varepsilon)) \subseteq \delta_i(\varepsilon)$ for $i = 1,2$.

Explicit computation of $T(\varepsilon)$ can be difficult.

The functional calculus enables one to get a decomposition like (3) but in terms of contour integrals which more directly show the relationship of $C_1(\varepsilon)$, $C_2(\varepsilon)$, $T(\varepsilon)$ to $C(\varepsilon)$. The essential features of this theory which we shall need are as follows.

If C is an $n\times n$ matrix and $f(\lambda)$ is an analytic function defined on $\sigma(C)$, then

$$f(C) = \frac{1}{2\pi i}\int_{\mathcal{C}} f(\lambda)(\lambda - C)^{-1}d\lambda \tag{4}$$

where $\mathcal{C}$ is a contour (nice closed curve) around $\sigma(C)$. Suppose that $\sigma(C) = \delta_1 \cup \delta_2$ where δ_1, δ_2 are two disjoint, nonempty sets. Let $\mathcal{C}$ be a contour with δ_1 on the inside and δ_2 on the outside. Then

$$P_1 = \frac{1}{2\pi i}\int_{\mathcal{C}} (\lambda - C)^{-1}d\lambda \tag{5}$$

is a projection ($P_1^2 = P_1$) that commutes with C. Thus

$$C = CP_1 + C(I - P_1) \tag{6}$$

and $\sigma(CP_1) \subseteq \delta_1 \cup \{0\}$, $\sigma(C(I - P_1)) \subseteq \delta_2 \cup \{0\}$. The decomposition (6) is the same as (3) written as

$$C = T\begin{bmatrix} C_1 & 0 \\ 0 & 0 \end{bmatrix}T^{-1} + T\begin{bmatrix} 0 & 0 \\ 0 & C_2 \end{bmatrix}T^{-1}. \tag{7}$$

Since $P_1C = CP_1$, P_1 also breaks down (decomposes) any function of C. In particular,

$$e^{C}P_1 = e^{CP_1}P_1 = \frac{1}{2\pi i}\int_C e^{\lambda}(\lambda - C)^{-1}d\lambda,$$

so that

$$e^{C} = e^{C}P_1 + e^{C}(I - P_1) = e^{P_1CP_1}P_1 + e^{(I-P_1)C(I-P_1)}(I - P_1). \quad (8)$$

Formula (8) will play a key role in what follows.

OUTLINE OF TECHNIQUE

In outline the technique proceeds as follows. Though it is not essential, (see [7]) assume that $A(\varepsilon)$ in (1) is invertible for $\varepsilon > 0$ but not for $\varepsilon = 0$. Basic to understanding the behavior of solutions of (1), (2), is an analysis of $\exp(A(\varepsilon)^{-1}B(\varepsilon)t)$ for small $\varepsilon \geq 0$. The eigenvalues of $A(\varepsilon)^{-1}B(\varepsilon)$ viewed as (possibly discontinuous) function of ε fall into one of four sets $\delta_1(\varepsilon)$, $\delta_2(\varepsilon)$, $\delta_3(\varepsilon)$, $\delta_4(\varepsilon)$ where $\delta_1(\varepsilon)$ is bounded, $\delta_2(\varepsilon)$ have real part $\to +\infty$, $\delta_3(\varepsilon)$ have real part go to $-\infty$, and $\delta_4(\varepsilon)$ have real part bounded and imaginary part unbounded. Using contour integrals of the resolvent, $(\lambda - A(\varepsilon)^{-1}B(\varepsilon))^{-1}$, around these groups of eigenvalues it is usually possible to get, as in (8)

$$\exp(A(\varepsilon)^{-1}B(\varepsilon)t) = \sum_{i=1}^{4} \exp(F_i(\varepsilon)t)P_i(\varepsilon) \quad (9)$$

where the $P_i(\varepsilon)$ are commuting projections and $\sum_{i=1}^{4} P_i(\varepsilon) = I$. Each $P_i(\varepsilon)$ is analytic in ε at zero, and each $F_i(\varepsilon)$ has one of the four groups of eigenvalues of $A(\varepsilon)^{-1}B(\varepsilon)$ as its nonzero eigenvalues. See, for example, [16]. In many problems it is possible to characterize the $\delta_i(\varepsilon)$. See, for example, [4], [6], [10].

Often one can determine the order of the pole of $F_i(\varepsilon)$, say m_i. Then letting $\tau_i = t\varepsilon^{-m_i}$ we get that

$$\exp(F_i(\varepsilon)t)P_i(\varepsilon) = \exp(\varepsilon^{m_i}F_i(\varepsilon)\tau_i)P_i(\varepsilon) = \sum_{n=0}^{\infty} \varepsilon^{n}X_{ni}(\tau_i) \quad (10)$$

since $\varepsilon^{m_i}F_i(\varepsilon)$ is analytic in ε. Note that the series actually converges and each $\exp(F_i(\varepsilon)t)P_i(\varepsilon)$ is a solution of (1). Each series in (10) is then inserted back into (1)-(2) and the $X_{ni}(\tau_i)$ solved for. The approach used will depend on the nature of $A(\varepsilon)$, $B(\varepsilon)$. Sometimes the $X_{ni}(\tau_i)$ are solved iteratively in the standard way. Sometimes the resulting system of ordinary differential equations is singular and the results of [8],[6] must be used. In other cases, the resulting system must be treated as an infinite system in order to solve for the $X_{ni}(\tau_i)$. See [6, Section 5.7]

In any case at least some of the terms can also be computed by contour integrals.

Even when solved iteratively the contour integrals help since from (10),

$$\sum_{n=0}^{\infty} \varepsilon^n X_{ni}(0) = P_i(\varepsilon) = \frac{1}{2\pi i}\int_{C_i} (\lambda - A(\varepsilon)^{-1}B(\varepsilon))^{-1} d\lambda$$

$$= \frac{1}{2\pi i}\int_{C_i} (\lambda A(\varepsilon) - B(\varepsilon))^{-1} A(\varepsilon) d\lambda \ .$$

Thus the expansions for $P_i(\varepsilon)$ provide all the necessary initial conditions.

UNIQUE FEATURES

Since the $P_i(\varepsilon)$ are analytic in ε, the solution $\underline{x}_\varepsilon(t)$ of (1) converges if and only if each $P_i(\varepsilon)\underline{x}_\varepsilon(t)$ does. Thus if $\underline{x}_\varepsilon(0)$ is the initial condition for (1), it is possible to completely characterize when and in what sense $\underline{x}_\varepsilon(t)$ converges to a solution of the reduced problem. Of course, depending on the $A(\varepsilon)$, $B(\varepsilon)$, $\underline{f}(\varepsilon,t)$ this could be computationally difficult. Boundary value problems could also be approached by this method.

This approach has certain unique features to it. The solution is broken into a sum of solutions each of whose values lie in a different subspace which depends on ε. Each of these solutions is analytic in ε and actually converges. One of these solutions goes uniformly on compact intervals of $[0,\infty)$ to a solution of the reduced problem.

Note that the slow and fast series are both defined for all t. Rather than having solutions which must match up this approach breaks the solution up into a nice solution and various slow and fast terms all of which are defined for all t but are only locally significant.

EXAMPLES

As a very simple but illustrative example, consider

$$\begin{aligned} \dot{x} &= x + \varepsilon y \\ \varepsilon\dot{y} &= -y \end{aligned} \tag{11}$$

(11) is easily solved directly as

$$x(t) = \frac{-\varepsilon^2}{\varepsilon+1} e^{-t/\varepsilon} y(0) + e^t\left(x(0) + \frac{\varepsilon^2}{\varepsilon+1} y(0)\right), y(t) = e^{-t/\varepsilon} y(0).$$

Now (11) may be written as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} 1 & \varepsilon \\ 0 & -1/\varepsilon \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} .$$

Let $C(\varepsilon) = \begin{bmatrix} 1 & \varepsilon \\ 0 & -1/\varepsilon \end{bmatrix}$. Then $\sigma(C(\varepsilon)) = \{1\} \cup \{-1/\varepsilon\}$. Let C_1 be a contour around 1 but not $-1/\varepsilon$. Then

$$P_1(\varepsilon) = \frac{1}{2\pi i} \int_{C_1} \begin{bmatrix} \lambda-1 & -\varepsilon \\ 0 & +1/\varepsilon \end{bmatrix}^{-1} d\lambda$$

$$= \frac{1}{2\pi i} \int_{C_1} \begin{bmatrix} (\lambda-1)^{-1} & \varepsilon(\lambda-1)^{-1}(\lambda+1/\varepsilon)^{-1} \\ 0 & (\lambda+1/\varepsilon)^{-1} \end{bmatrix} d\lambda = \begin{bmatrix} 1 & \varepsilon^2/1+\varepsilon \\ 0 & 0 \end{bmatrix} .$$

Thus

$$e^{F_1(\varepsilon)t} P_1(\varepsilon) = e^{\begin{bmatrix} 1 & \varepsilon^2/1+\varepsilon \\ 0 & 0 \end{bmatrix} t} \begin{bmatrix} 1 & \varepsilon^2/1+\varepsilon \\ 0 & 0 \end{bmatrix} = e^t \begin{bmatrix} 1 & \varepsilon^2/1+\varepsilon \\ 0 & 0 \end{bmatrix} \tag{12}$$

whereas

$$e^{\varepsilon F_2(\varepsilon)\tau} P_2(\varepsilon) = e^{\varepsilon \begin{bmatrix} 1 & \varepsilon \\ 0 & -1/\varepsilon \end{bmatrix} \begin{bmatrix} 0 & -\varepsilon^2/1+\varepsilon \\ 0 & 1 \end{bmatrix} \tau} \begin{bmatrix} 0 & -\varepsilon^2/1+\varepsilon \\ 0 & 1 \end{bmatrix}$$

$$= e^{\begin{bmatrix} 0 & \varepsilon^2/1+\varepsilon \\ 0 & -1 \end{bmatrix} \tau} \begin{bmatrix} 0 & -\varepsilon^2/1+\varepsilon \\ 0 & 1 \end{bmatrix} \tag{13}$$

$$= e^{-\tau} \begin{bmatrix} 0 & -\varepsilon^2/1+\varepsilon \\ 0 & 1 \end{bmatrix}$$

We then arrive at the solution

$$\begin{bmatrix} x_\varepsilon \\ y_\varepsilon \end{bmatrix} = e^t \begin{bmatrix} 1 & \varepsilon^2/1+\varepsilon \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_\varepsilon(0) \\ y_\varepsilon(0) \end{bmatrix} + e^{-t/\varepsilon} \begin{bmatrix} 0 & -\varepsilon^2/1+\varepsilon \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_\varepsilon(0) \\ y_\varepsilon(0) \end{bmatrix} . \tag{14}$$

Note that the solution (14) is composed of two solutions (12), (13), each of which exists for all t. (12) lies in the one-dimensional subspace spanned by the vector $[1,0]^T$ and converges uniformly on $[0,\infty)$ to the solution of the reduced problem $x = e^t x(0)$, $y = 0$. (13) lies in the one dimensional subspace spanned by $[\varepsilon^2(1+\varepsilon)^{-1}, -1]^T$ and is the boundary correction. It is significant only for "small" values of $\tau = t/\varepsilon$.

In [6] we generally talked about (1)-(2) with $\underline{x}_\varepsilon(0)$ independent of ε. However, our approach makes it easy to characterize when $\underline{x}_\varepsilon(t)$ converges to a solution of the reduced problem.

To illustrate this point consider the example. Suppose that $\underline{x}_\varepsilon(0)$ is a given function of $\varepsilon > 0$. Let $\underline{x}_\varepsilon(t)$ be the solution of (1). Then $\underline{x}_\varepsilon(t)$ will converge to a solution of the reduced problem if and only if

$$P_1(\varepsilon)\underline{x}_\varepsilon(0) \quad \text{converges} \tag{15}$$

and

$$P_2(\varepsilon)\underline{x}_\varepsilon(0) \quad \text{converges to zero.} \tag{16}$$

That is,

$$x_\varepsilon(0) + \varepsilon^2 y_\varepsilon(0)(1+\varepsilon)^{-1} \quad \text{converges} \tag{17}$$

and

$$\begin{bmatrix} \varepsilon^2(1+\varepsilon)^{-1} y_\varepsilon(0) \\ -y_\varepsilon(0) \end{bmatrix} \to 0,$$

or equivalently

$$y_\varepsilon(0) \to 0. \tag{18}$$

Of course, (11) can be easily analyzed directly but this example illustrates how our approach can be used to analyze such problems.

The second example is from [7]. Consider

$$\begin{bmatrix} 0 & 1 \\ \varepsilon & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Let $A(\varepsilon) = \begin{bmatrix} 0 & 1 \\ \varepsilon & 0 \end{bmatrix}$. Then $\sigma(A(\varepsilon)) = \{\sqrt{\varepsilon}\} \cup \{-\sqrt{\varepsilon}\}$. Let $C(\varepsilon)$ be a small contour around $\sqrt{\varepsilon}$ but not including $-\sqrt{\varepsilon}$. Let

$$P_1(\varepsilon) = \frac{1}{2\pi i}\int_{C(\varepsilon)} (\lambda - A(\varepsilon))^{-1}\,d\lambda$$

$$= \frac{1}{2\pi i}\int_{C(\varepsilon)} \begin{bmatrix} \lambda & -1 \\ -\varepsilon & \lambda \end{bmatrix}^{-1} d\lambda = \frac{1}{2\pi i}\int_{C(\varepsilon)} \frac{1}{\lambda^2-\varepsilon}\begin{bmatrix} \lambda & 1 \\ \varepsilon & \lambda \end{bmatrix} d\lambda$$

$$= \frac{1}{2\sqrt{\varepsilon}}\begin{bmatrix} \sqrt{\varepsilon} & 1 \\ \varepsilon & \sqrt{\varepsilon} \end{bmatrix}.$$

Then $e^{-A^{-1}(\varepsilon)t} = e^{-t/\sqrt{\varepsilon}}P_1(\varepsilon) + e^{t/\sqrt{\varepsilon}}(I - P_1(\varepsilon))$.

$$= \frac{e^{-t/\sqrt{\varepsilon}}}{2\sqrt{\varepsilon}}\begin{bmatrix} \sqrt{\varepsilon} & 1 \\ \varepsilon & \sqrt{\varepsilon} \end{bmatrix} + \frac{e^{t/\sqrt{\varepsilon}}}{2\sqrt{\varepsilon}}\begin{bmatrix} \sqrt{\varepsilon} & -1 \\ -\varepsilon & \sqrt{\varepsilon} \end{bmatrix}.$$

Of course, this example could easily be done by computing eigenvectors corresponding to $\pm\sqrt{\varepsilon}$ but the above procedure works even if there is a deficiency of eigenvectors.

DISCUSSION

While this approach has some nice points it has some limitations. As developed in [6] it is linear and autonomous. However, linear autonomous problems serve to suggest possibilities for nonlinear, nonautonomous ones. It can be modified to handle infinite dimensional systems and hence certain partial differential equations but it is heavily linear.

The extension of these results to infinite dimensional problems is natural, since that is where the machinery was originally developed in [16]. Do the results actually provide insight on singularly perturbed partial differential equations? That remains to be determined. Some work on extending the results of [3], [9] to problems involving certain partial differential equations may be found in [12], [13].

The recent paper by O'Malley [24] on nonautonomous systems while not done using contour integrals, involved decompositions and assumptions that resemble those of the autonomous case.

For nonlinear equations, the analogue of e^{At} involves semigroups. Can one carry over at least the flavor of this approach to that setting? It looks hard.

REFERENCES

1. D. J. Bell and D. H. Jacobson, Singular Optimal Control Problems, Academic Press, New York, 1975.

2. V. F. Butuzov and A. B. Vasiléva, Differential and difference equation systems with a small parameter for the case in which the unperturbed (singular) system is in the spectrum, Diff. Eqn. 6(1970), 499-510.

3. S. L. Campbell, Optimal control of autonomous linear processes with singular matrices in the quadratic cost functional, SIAM J. Control 14(1976), 1092-1106.

4. S. L. Campbell, Singular perturbation of autonomous linear systems, II, Jrnl. Diff. Eqn., 29(1978), 362-373.

5. S. L. Campbell, On a singularly perturbed autonomous linear control problem, IEEE. Trans. Automatic Control AC-24, (1979), 115-116.

6. S. L. Campbell, Singular Systems of Differential Equations, Pitman Publishing Co., 1979 (Dec.) (In press).

7. S. L. Campbell, A more singular singularly perturbed linear system, preprint.

8. S. L. Campbell and C. D. Meyer, Jr., Generalized Inverses of Linear Transformations, Pitman Publishing Co., Ltd., 1979.

9. S. L. Campbell, C. D. Meyer, Jr., and N. J. Rose, Applications of the Drazin inverse to linear systems of differential equations, SIAM J. Appl. Math., 31(1976), 411-425.

10. S. L. Campbell and N. J. Rose, Singular Perturbation of Autonomous Linear Systems, SIAM J. Math. Anal., 10(1979), 542-551.

11. S. L. Campbell and N. J. Rose, Singular perturbation of autonomous linear systems III, Houston, J. Math 4(1978), 527-539.

12. A. Favini, Laplace transform method for a class of degenerate evolution problems, Rend. Mat. (Roma), 1979 (to appear).

13. A. Favini, Degenerate evolution equations and singular optimal control, preprint, 1979.

14. W. A. Harris, Jr., Singularly perturbed boundary value problems revisited, Symposium on Ordinary Differential Equations, Lecture Notes in Mathematics, No. 312, Springer-Verlag, New York, 1972, 54-64.

15. D. H. Jacobson, Totally singular quadratic minimization problems, IEEE Trans. Automatic Control, 16(1971), 651-657.

16. T. Kato, Perturbation Theory for Linear Operators, Springer-Verlag, New York, 1966.

17. P. K. Kokotovic, Singular perturbations in optimal control, Rocky Mtn. J. Math. 6(1976), 767-773.

18. P. K. Kokotovic and P. Sannuti, Singular perturbation method for reducing the model order in optimal control design, IEEE Trans. Automatic Control 13(1968), 377-384.

19. P. K. Kokotovic and R. A. Yuckel, Singular perturbation theory of linear state regulators, IEEE Trans. Automatic Control, 17(1972), 29-37.

20. G. A. Kurina, Asymptotic solution of a classical singularly perturbed optimal control problems, Soviet Math. Dok. 18(1977), 722-726.

21. R. E. O'Malley, Jr. and A. Jameson, Singular perturbations and singular arcs - Part I, IEEE Trans. Automatic Control AC-20 1975, 218-226.

22. R. E. O'Malley, Jr., Cheap control, singular arcs, and singular perturbations, Lecture Notes in Economics and Mathematics Systems, No. 106, Springer-Verlag.

23. R. E. O'Malley, Jr. and J. E. Fhaherty, Singular Singular-Perturbation problems, Lecture Notes in Mathematics, No. 594, Springer-Verlag, 422-436.

24. R. E. O'Malley, Jr., A singular singularly-perturbed linear boundary value problem, SIAM J. Math. Anal., 10(1979), 695-708.

25. A. B. Vasiléva, Singularly perturbed systems containing indeterminacy in the case of degeneracy, Soviet Math. Dokl. 16(1975), 1121-1125.

26. R. R. Wilde and P. K. Kokotovic, Optimal open and closed-loop control of singularly perturbed linear system, IEEE Trans. Automatic Control, AC-18, (1973), 616-626.

27. K. D. Young, P. K. Kokotovic, V. I. Utkin, A singular perturbation analysis of high-gain feedback systems, IEEE Trans. Automatic Control, AC-22, (1977), 931-938.

STABILIZATION OF TIME-VARYING SINGULARLY PERTURBED SYSTEMS BY OBSERVER-BASED SLOW-STATE FEEDBACK

S. HAROLD JAVID
Systems Control, Inc., Palo Alto, CA 94304

ABSTRACT

This paper develops an observer based on the slow reduced-order system of a time-varying full-order system. An upper bound μ^* on the singular perturbation parameter μ is derived such that for μ smaller than μ^*, the full-order system is stabilized by linear feedback of the observed slow state. The dependence of this bound on the system matrices and the observer and controller gain matrices is discussed.

I. INTRODUCTION

A common simplification used in developing control and estimation algorithms is the reduction of system order by neglecting "fast" sensor and actuator dynamics, "parasitics." One model which may be used to represent a system with "slow" plant dynamics and "fast" sensor and actuator dynamics is

$$\dot{x} = A_{11}(t)x + A_{12}(t)z + B_1(t)u \tag{1a}$$

$$\mu\dot{z} = A_{21}(t)x + A_{22}(t)z + B_2(t)u \tag{1b}$$

$$y = C_1(t)x + C_2(t)z + D(t)u \tag{1c}$$

where $x \varepsilon R^n$, $z \varepsilon R^m$, $u \varepsilon R^r$, and $y \varepsilon R^p$ and μ is a small positive scalar. The states x and z represent the plant dynamic state (slow) and the sensor and actuator states (fast); u and y are the control and measurement vectors. The elements of the full system (1) matrices are assumed to be of the same order of magnitude. Consequently, for specified x, z and u, $\dot{z}$ is on the order of $1/\mu$ time $\dot{x}$. For small μ, $\dot{z}$ is much greater than $\dot{x}$. As a result we say that z is a fast state and x is a slow state.

The reduced system

$$\dot{\overline{x}} = A_o(t)\overline{x} + B_o(t)u \tag{2a}$$

$$\overline{y} = C_o(t)\overline{x} + D_o(t)u \tag{2b}$$

where*

$$A_o = A_{11} - A_{12}A_{22}^{-1}A_{21} \tag{3a}$$

$$B_o = B_1 - A_{12}A_{22}^{-1}B_2 \tag{3b}$$

$$C_o = C_1 - C_2A_{22}^{-1}A_{21} \tag{3c}$$

$$D_o = D - C_2A_{22}^{-1}B_2 \tag{3d}$$

---

*All the matrices in this paper are time varying. To simplify the notation the dependence on t is not always shown.

is obtained by setting $\mu=0$ and assuming $A_{22}^{-1}(t)$ exists. We may assume that (2) is the model of (1) which would be obtained by neglecting fast sensor and actuator dynamics. An observer which would be designed based on the assumption that (2) was an adequate model of (1) takes the form

$$\dot{\hat{x}} = A_o\hat{x} + B_o u + K_o(t)(y-\hat{y}) \tag{4a}$$

$$\hat{y} = C_o\hat{x} + D_o u. \tag{4b}$$

The designer of such an observer would intend that $\hat{x}$ be an approximation of x, the slow state of the full system (1). By a simple extension of the results in [1], it can be shown that $e(t)\triangleq x-\hat{x}$ will satisfy

$$e(t) = \Phi_o(t,t_o)e(t_o) + \mu W(z,x,u,t) \tag{5}$$

where Φ_o is the state transition matrix of

$$\dot{r} = (A_o - K_oC_o)r \tag{6}$$

and W is bounded. This suggests that $K_o(t)$ be chosen so that (6) is uniformly asymptotically stable.

Suppose that a feedback,

$$u = K(t)\hat{x}, \tag{7}$$

is applied to the full system where K(t) is chosen so that the equilibrium w=0 of

$$\dot{w} = [A_o(t) + B_o(t)K(t)]w \tag{8}$$

is uniformly asymptotically stable. Then, if the eigenvalues of $A_{22}(t)$ have real parts $\mathrm{Re}(\lambda_i(t))$ which satisfy

$$\mathrm{Re}(\lambda_i(t)) \leq -\gamma < 0 \tag{9}$$

where γ is constant, there exists a μ^* such that for $\mu\varepsilon(o,\mu^*)$ the full system is uniformly asymptotically stable. A bound on μ is the main result of this paper. Such an approach to stabilization has the advantage that the observer is of order n which is m less than the full system.

A similar result is derived in [2,3] for time-invariant singularly perturbed systems. In these papers no bound on μ is given; only the existence of μ^* is proven. Thus this paper extends the results of [2,3] in two ways. First, it treats time-varying systems. Second, it derives a bound which is valid both for time-varying and time-invariant systems.

Section II of this paper summarizes the results of [4] which provide the bound and Section III contains the main result of this paper. Conclusions are given in Section IV.

II. STABILITY

The results of [4] summarized in this section will be used to derive the main result of this paper in the next section.

$$\dot{\bar{v}} = [E_{11}(t) - E_{12}(t)E_{22}^{-1}(t)E_{21}(t)]\,\bar{v}_1 \triangleq E_1(t)\bar{v}_1, \bar{v}_1(t_o) = v_{10} \tag{10}$$

is a singular perturbation of

$$\dot{v}_1 = E_{11}(t)v_1 + E_{12}(t)v_2 \qquad v_1(t_o) = v_{10} \tag{11a}$$

$$\mu\dot{v}_2 = E_{21}(t)v_1 + E_{22}(t)v_2 \qquad v_2(t_o) = v_{20} \tag{11b}$$

which is obtained by setting $\mu=0$.

Assumption 1: The matrices $E_{ij}(t)$ $i,j=1,2$ are bounded and have bounded derivatives for all t.

Assumption 2: The eigenvalues of $E_{22}(t)$ satisfy

$$\mathrm{Re}(\lambda_i(t)) < -\gamma < 0 \tag{12}$$

for all t where γ is constant.

Under these assumptions, this section gives a bound μ^* such that for $\mu\varepsilon(0,\mu^*)$, system (11) is uniformly asymptotically stable.

A condition for $\bar{v}_1(t)$ to be an $O(\mu)$ approximation of $v_1(t)$ is the uniformly asymptotic stability of the fast subsystem

$$\mu\dot{\tilde{v}}_2 = E_{22}(t)\tilde{v}_2 \tag{13}$$

Thus it is of interest to determine the stability of (13). Clearly, Assumptions 1 and 2 are not enough for (13) to be stable.

Assumption 1 implies that

$$|E_{22}(t) - E_{22}(t_o)| \le \beta(t-t_o) \tag{14}$$

where, by the mean value theorem, β is a positive constant equal to the maximum of $|\dot{A}_{22}(t)|$ for all t. Also, there exists k such that

$$\left|\exp\left(A_{22}(t_o)\left(\frac{t-t_o}{\mu}\right)\right)\right| \le k \exp\left(-\gamma\left(\frac{t-t_o}{\mu}\right)\right) \tag{15}$$

under Assumption 2.

Theorem I: Let Assumptions 1 and 2 be satisfied and $\mu_o = \gamma^2/\beta k$. Then for $\mu\varepsilon(0,\mu_o)$, system (13) is uniformly asymptotically stable.

Theorem II: Let (10) and (13) be uniformly asymptotically stable systems so that their state transition matrices satisfy (16) and (17) respectively

$$|\Phi_1(t,t_o)| \le k_1 \exp(-\sigma_1(t-t_o)) \tag{16}$$

$$|\Phi_{22}(t-t_o)| \le k_2 \exp\left(-\sigma_2\left(\frac{t-t_o}{\mu}\right)\right) \tag{17}$$

Let $L(t) = E_{22}^{-1}(t)E_{21}(t)$, M_1,M_2 and M_3 exist such that for all t

$$|E_{12}(t)| \le M_1, |L(t)E_{12}(t)| \le M_2, |\dot{L}(t) + L(t)E_1(t)| \le M_3$$

and

$$\mu_1 = \sigma_1\sigma_2/(\sigma_1 k_2 M_2 + k_1 M_1 k_2 M_3) \ .$$

Then for $\mu\varepsilon(0,\mu_1)$, the full system (11) is uniformly asymptotically stable.

The next corollary follows immediately from Theorems I and II.

Corollary: Let $\mu^* = \min(\mu_0,\mu_1)$. If the slow subsystem (10) is uniformly asymptotically stable, then Assumptions 1 and 2 guarantee for $\mu\varepsilon(0,\mu^*)$, system (1) is uniformly asumptotically stable.

The proofs of Theorems I and II are in [4] along with a discussion of how these stability results relate to other stability results for singularly perturbed systems. In the next section, these results are applied to the problem of stabilizing the full system (1) by reduced-observer slow-state feedback.

III. REDUCED-OBSERVER SLOW-STATE STABILIZATION

In Section I the stabilization problem treated in this paper is introduced. Section II presents the stability results needed to establish the results presented in this section. A few general comments are in order before presenting these results.

The state $\bar{x}$ of slow reduced system (2) resulting from setting μ=0 in the full system (1) is an approximation of x in system only when the fast dynamics possess certain stability properties. The quasi-equilibrium

$$0 = A_{21}(t)\bar{x} + A_{22}(t)\bar{z} + B_2(t)u \tag{18}$$

must be stable in the sense that for a continuous control u(t) t must approach to within 0(μ) of

$$\bar{z} = -A_{22}^{-1}A_{21}\bar{x} - A_{22}^{-1}B_2 u \tag{19}$$

on an 0(μ) interval. A sufficient condition for this is the uniform asymptotic stability of the fast subsystem

$$\mu\dot{\tilde{z}} = A_{22}(t)\tilde{z} \ . \tag{20}$$

Theorem 1 provides a bound for this uniform asymptotic stability. Since this stability is necessary for $\bar{x}$ to approximate x, it is also necessary for the reduced-observer state $\hat{x}$ to approach to with 0(μ) of x. Of course $\hat{x}$ will not converge to an 0(μ) neighborhood in an 0(μ) time interval. It's convergence is dependent on $A_o(t) - K_o(t)C_o(t)$ as shown in Equations (5) and (6).

The feedback (7) and observer (4) is to be used to stabilize the slow dynamics of the full system. If we define $e = x - \hat{x}$, substitute (7) into (1) and (4) and suffer through some tedious algebra, we obtain

$$\begin{bmatrix} \dot{x} \\ \dot{e} \\ \mu\dot{z} \end{bmatrix} = \begin{bmatrix} A_{11} + B_1K & -B_1K & A_{12} \\ M_{21} & M_{22} & A_{12} - K_0C_2 \\ A_{21} + B_2K & -B_2K & A_{22} \end{bmatrix} \begin{bmatrix} x \\ e \\ z \end{bmatrix} \tag{21a}$$

where

$$M_{22} = A_o - B_o K + K_o DK - K_o C_o - K_o D_o K - B_1 K \tag{21b}$$

$$M_{21} = A_{11} - K_o C_1 - M_{22} . \tag{21c}$$

In order to use Stability Theorems I and II, we define

$$v_1 = \begin{bmatrix} x \\ e \end{bmatrix}, \; v_2 = z \tag{22}$$

and partition (21) accordingly. Setting $\mu=0$, we obtain

$$\dot{\bar{v}}_1 = \begin{bmatrix} \dot{\bar{x}} \\ \dot{\bar{e}} \end{bmatrix} = \begin{bmatrix} A_o + B_o K & -B_o K \\ 0 & A_o - K_o C_o \end{bmatrix} \begin{bmatrix} \bar{x} \\ \bar{e} \end{bmatrix} \triangleq E_1 \bar{v}_1 . \tag{23}$$

It may be shown that choosing K_o and K so that (6) and (8) are uniformly asymptotically stable will provide the uniform asymptotic stability of (23). Thus the natural choices of K and K_o made based on the assumption that the reduced-system (2) is an adequate model of the full system (1) stabilizes (23). Stability Theorems 1 and 2 apply directly, where

$$E_{22}(t) = A_{22}(t) \tag{24a}$$

$$L(t) = A_{22}^{-1}[A_{21} + B_2 K \,\vdots\, -B_2 K] \tag{24b}$$

$$E_{12} = \begin{bmatrix} A_{12} \\ A_{12} - K_o C_2 \end{bmatrix} \tag{24c}$$

and E_1 is defined in (23). The constants $\gamma, \beta, k, k_1, k_2, \sigma_1, \sigma_2, M_1, M_2$ and M_3 may now be defined by Equations (21) through (24). As a result μ_0 and μ_1 may be defined from which the bound μ^* may be found.

Assumptions 1 and 2 guarantee that for $\mu \varepsilon (0,\mu^*)$ the full system is stabilized by the slow-state reduced-observer feedback.

It is interesting to examine the dependence of μ^* on the matrices $K_o(t)$ and $K(t)$, as for a given full system the value of μ is fixed. It would be desirable to understand how $K(t)$ and $K_o(t)$ could be chosen to set μ^* so that $\mu \varepsilon (0,\mu^*)$. In all practical cases, the fast subsystem (20) will be uniformly asymptotically stable. Otherwise, the reduced system (2) would not be considered an adequate model of (1) and the reduced-observer (4) would not be used as full state (slow and fast) would be required. For this case $\mu^* = \mu_1$ which is defined

$$\mu_1 = \sigma_1 \sigma_2 / (\sigma_1 k_2 M_2 + k_1 M_1 k_2 M_3).$$

If the system is such that μ in (1) is less than μ_1, we can be certain that (1) is stabilized by this feedback structure. Since Theorem I is only a necessary condition for stability, system (1) may still be stabilized by the slow-state feedback, since bounds such as μ_1 are dependent on matrix norms and tend to be conservative.

IV. CONCLUSION

The main result of this paper is the extension of observer designs based on slow reduced order system model to the time-varying case. This extension includes a bound μ^* which may be used to determine whether μ in a given system is small enough for $K_o(t)$ and $K(t)$ assure system stability. Due to the conservativeness of μ^*, system (1) may be stabilized even if the conditions of Theorems I and II are not satisfied.

REFERENCES

[1] S. H. Javid, "Observing the Slow States of a Singularly Perturbed System," Proceedings of the 18th IEEE CDC, Ft. Lauderdale, FLA, Dec. 12-14, 1979.

[2] M. J. Balas, "Observer Stabilization of Singularly Perturbed Systems," J. Guidance and Control, Vol. 1, No. 1, Jan.-Feb. 1978.

[3] B. Porter, "Singular Perturbation Methods in the Design of Observers and Stabilizing Feedback Controllers for Multivariable Linear Systems," Electronics Letters, Vol. 10, No. 23, Nov. 14, 1974.

[4] S. H. Javid, "Uniform Asymptotic Stability of Linear Time-Varying Singularly Perturbed Systems," Journal of the Franklin Institute, Vol. 305, No. 1, January 1978.

APPROXIMATION ALGORITHMS FOR SOME GRAPH AUGMENTATION PROBLEMS (Extended abstract)

GREG N. FREDERICKSON and JOSEPH JA'JA'
Computer Science Department, The Pennsylvania State University
University Park, Pennsylvania 16802

Abstract

Graph augmentation problems on a weighted graph involve determining a minimum-cost set of edges to add to a graph to satisfy a specified property, such as biconnectivity, bridge-connectivity or strong connectivity. It is shown that these augmentation problems are NP-complete even if the graph initially is connected, e.g., a tree. Approximation algorithms with constant worst-case bounds are presented.

1. Introduction

A number of graph problems can be viewed as augmentation problems: Given a complete graph $G = (V, E)$, a subgraph $G_0 = (V, E')$ and a cost function on E, find a set of edges $E'' \subseteq E - E'$ of minimum cost such that $(V, E' \cup E'')$ satisfies a specified property. For instance, if $E' = \emptyset$ and the property is graph connectivity, then the corresponding problem is that of finding a minimum-cost spanning tree on V. An analogous problem for directed graphs, in which the property is having vertices in V be reachable from a given vertex, involves finding a minimum-cost spanning arborescence. This problem can be solved in an efficient manner [1, 3, 15]. Further examples of augmentation problems are cited in [4].

We consider three properties studied by Eswaran and Tarjan [4]: strong connectivity for directed graphs, and biconnectivity and bridge-connectivity for undirected graphs. The latter properties may be viewed as certain desirable features of networks. Given a network, we wish to augment the network with additional edges so that these features are achieved. Networks that are biconnected and bridge-connected can survive single element failures: A biconnected network can survive a node failure, and a bridge-connected network can survive a communication line failure.

If the edges in the graph all have equal weight, then the various augmentation problems also have efficient algorithms [4], [14]. However, if the edges have unequal weights, then the augmentation problems are NP-complete [4]. The problems remain NP-complete if G_0 is restricted to be a connected subgraph, with edge weights chosen from the set $\{1, 2\}$, as we shall see in section 2. In particular, augmenting a directed acyclic graph to be strongly connected, or augmenting even a tree to be bridge-connected or biconnected, is NP-complete.

Membership in the NP-complete [2, 11, 12, 8] class of problems appears to indicate that there is no efficient algorithm for solving a problem exactly. A reasonable alternative is to design efficient algorithms that yield near-optimal, or approximate, solutions [10, 7]. In sections 3, 4, and 5 we present approximation algorithms for our augmentation problems, and bound their performance in terms of a worst case ratio of the cost of a generated augmentation to the cost of an optimal augmentation. Our algorithms run in $O(|V|^2)$ time, and have a similar strategy: Use an algorithm for finding a spanning arborescence to strongly connect a weakly connected graph. We consider graphs with no special cost restrictions, such as the triangle inequality, but note that improved bounds can be achieved for certain problems if the graphs are so restricted [6].

We note that our graph terminology is standard (see, for example, [9]). We only make the following definitions. An arborescence is a directed acyclic graph with one vertex, the root, having no entering edges, and all other vertices having exactly one entering edge. A branching of a directed graph G is a graph whose weakly connected components are arborescences.

2. NP-completeness Results

We state formal definitions of the bridge-connectivity, biconnectivity, and strong connectivity augmentation problems below, and then proceed to show that restricted version of these problems are NP-complete:

BRIDGE-CONNECTIVITY AUGMENTATION (BRA):

Instance: Complete undirected graph $G = (V, E)$, weight function $c(e) \varepsilon Z^+$ for $e \varepsilon E$, subgraph $G_0 = (V, E')$, $E' \subseteq E$, and positive integer B.

Question: Is there a set $E'' \subseteq E - E'$ such that $\sum_{e \varepsilon E''} c(e) \leq B$, and $(V, E' \cup E'')$ is bridge-connected?

BICONNECTIVITY AUGMENTATION (BIA):

Same as BRA except with biconnected replacing bridge-connected.

STRONG CONNECTIVITY AUGMENTATION (STA):

Instance: Complete directed graph $G = (V, A)$, weight function $c(a) \varepsilon Z^+$ for $a \varepsilon A$, subgraph $G_0 = (V, A')$, with $A' \subseteq A$, and positive integer B.

Question: Is there a set $A'' \subseteq A - A'$ such that $\sum_{a \varepsilon A''} c(a) \leq B$, and $(V, A' \cup A'')$ is strongly connected?

Eswaran and Tarjan [4] have shown all three augmentation problems to be NP-complete. Their reductions are from Hamiltonian circuit, and choose G_0 to be the empty subgraph $(V, \emptyset)$. It is possible that the problems could be easier if they are restricted to a connected G_0, in particular a tree. Let the connected bridge-connectivity augmentation problem (CBRA) be the same as BRA, except that G_0 must be a connected subgraph of G. Let the connected biconnectivity augmentation problem (CBIA) be similarly defined. Let the connected strong-connectivity augmentation problem (CSTA) be the same as STA, except that G_0 must be weakly connected. We show that all three restricted problems, CBRA, CBIA, and CSTA, are NP-complete.

Eswaran and Tarjan [4] claimed, but did not provide proofs, that CBIA and CBRA are NP-complete. Tarjan [16] has provided a proof only for CBIA, using a transformation from directed Hamiltonian circuit. It appears that his transformation for CBIA does not carry over to CBRA.

Our transformation for CBRA is from 3-dimensional matching:

3-DIMENSIONAL MATCHING (3DM):

Instance: A set $M \subseteq W \times X \times Y$, where W, X and Y are disjoint sets having the same number q of elements.

Question: Does M contain a matching, that is, a subset $M' \subseteq M$ such that $|M'| = q$ and no two elements of M' agree in any coordinate?

The transformations for CSTA and CBIA are also from 3DM, and are essentially the same. All three restricted augmentation problems remain

NP-complete if edge weights are either 1 or 2. Handling G_0 where G_0 is a tree is no easier, since our reductions are to a graph that is a tree. Since it is easy to see that all three problems are in NP, we prove that 3DM is reducible to each of our problems, e.g., 3DM α CBRA. We provide one of the proofs below.

Theorem 1. 3DM α CBRA.

Proof. Let $M \subseteq W \times X \times Y$ be an instance of 3DM, with $|M| = p$ and $W = \{w_i | i = 1, 2, \ldots, q\}$, $X = \{x_i | i = 1, \ldots, q\}$ and $Y = \{y_i | i = 1, \ldots, q\}$. We define an instance of CBRA as follows:

$$V = \{r\} \cup \{w_i, x_i, y_i | i = 1, \ldots, q\}$$
$$\cup \{a_{ijk}, \bar{a}_{ijk} | (w_i, x_j, y_k) \varepsilon M\}$$

$$E = \{(u, v) | u, v \varepsilon V \text{ and } u \neq v\}$$

$$E' = \{(r, w_i), (r, x_i), (y_i, r) | i = 1, \ldots, q\}$$
$$\cup \{(a_{ijk}, w_i), (w_i, \bar{a}_{ijk}) | (w_i, x_j, y_k) \varepsilon M\}$$

Let $c(\bar{a}_{ijk}, a_{ijk}) = c(x_j, a_{ijk}) = c(\bar{a}_{ijk}, y_k) = 1$, for $(w_i, x_j, y_k) \varepsilon M$. Let all other edges in E have weight 2. Let $B = p + q$.

We claim that M contains a matching M' iff there is a set E'' of cost no more than $B = p + q$, such that $(V, E' \cup E'')$ is bridge-connected. Suppose M contains a matching M'. For each triple (w_i, x_j, y_k) in M', insert (x_j, a_{ijk}) and $(\bar{a}_{ijk}, y_k)$ into E''. This will cause w_i, x_j, and y_k to be on cycle in $E' \cup E''$ containing r. Since $|M'| = q$, the total cost will be $2q$, and all w_i, x_j, and y_k will be on cycles containing r.

For each triple (w_i, x_j, y_k) in $M - M'$, insert $(\bar{a}_{ijk}, a_{ijk})$ into E''. This will cause each a_{ijk} and $\bar{a}_{ijk}$ not already on a cycle in $E' \cup E''$ to be on a cycle containing w_i. Total cost of this step will be $|M| - |M'| = p - q$. Hence the total cost of E'' will be $p + q$. Since E' is connected, and every vertex in V is on a cycle, the removal of any one edge will not disconnect the graph. Hence $(V, E' \cup E'')$ is bridge-connected.

We now prove the "if" part: Assume that there is a set of E'' of cost no greater than $p + q$, such that $(V, E' \cup E'')$ is bridge-connected. There are $2(p + q)$ leaves in (V, E'). For $(V, E' \cup E'')$ to be bridge-connected, E'' must contain at least $p + q$ edges, since an edge can connect at most two leaves. Since E'' is of cost $p + q$, E'' contains exactly $p + q$ edges, each of cost 1. Hence each leaf in (V, E') has exactly one edge of E'' incident on it.

Each leaf x_j will have one edge (x_j, a_{ijk}) from E'' incident on it. Hence $(\bar{a}_{ijk}, a_{ijk})$ is not in E'', and thus $(\bar{a}_{ijk}, y_k)$ is in E''. No other edge in E'' will be incident to y_k. Hence for each x_j, the corresponding (w_i, x_j, y_k) will be in M'. The remaining $p - q$ edges in E'' will be of the form $(\bar{a}_{ijk}, a_{ijk})$, and thus contribute nothing to the matching. □

3. Strong Connectivity Augmentation Algorithm

We now present an algorithm that will find a set of directed edges that makes a subgraph strongly connected. Our strategy is to add edges to the augmenting set A'' in two stages. In the first stage we find a reverse spanning arborescence, i.e., a directed acyclic spanning subgraph with one vertex, the root, having no leaving edge, and all other vertices having exactly one leaving edge. There is a path from every vertex to the root in such a directed spanning tree. We use an algorithm for finding optimum branchings [3, 1, 15] to find the reverse spanning arborescence.

In the second stage, we find a spanning arborescence, forcing the root to be the same as was returned from the first stage. In this directed spanning tree, all vertices have exactly one entering edge, except the root, which has none. We can force a particular vertex to be the root by setting costs on all entering edges to be very large. Hence there will be a path from the root to every vertex. Combining the two directed spanning trees will yield a graph in which there is a path from any vertex u to the root and a path from the root to any vertex v. Thus the graph will be strongly connected.

We note that our algorithm generates an optimal solution in the case that either a spanning arborescence or a reverse spanning arborescence is contained in G_0. This is essentially the same observation made in [4] with respect to one special case that they observed was efficiently solvable.

Algorithm STC

Input: $G = (V, A)$ a complete directed graph with cost function $c: A \to Z^+$, and $G_0 = (V, A')$, a subgraph of G.

Output: A set of edges $A'' \subset A - A'$ such that $(V, A' \cup A'')$ is strongly connected.

1. For each edge $\langle u, v\rangle$ in A, let $d\langle u, v\rangle = 0$ if $\langle v, u\rangle \varepsilon A'$ and let $d\langle u, v\rangle = c\langle v, u\rangle$ otherwise.
2. Find a minimum weight spanning arborescence $T = (V, A_a')$ with root $r \varepsilon V$ on (V, A) using d. Set $A_a = \{\langle u, v\rangle | \langle v, u\rangle \varepsilon A_a'\}$.
3. For each edge $\langle u, v\rangle$ in A, let $d'\langle u, v\rangle = c\langle u, v\rangle$. If $\langle u, v\rangle \varepsilon A' \cup A_a$, then set $d'\langle u, v\rangle = 0$. For all u, set $d'\langle u, r\rangle = \infty$.
4. Find a minimum weight spanning arborescence $T' = (V, A_b)$ on (V, A) using d'.
5. Set $A'' = (A_a \cup A_b) - A'$. □

Lemma 1. Let $G = (V, A)$ be a complete directed graph with cost function $c: A \to Z^+$ and let $G_0 = (V, A')$ be a subgraph of G. Then algorithm STC finds a set of edges $A'' \subset A - A'$ such that $(V, A' \cup A'')$ is strongly connected.

The proof is straightforward and will be omitted here.

Theorem 2. Let $G = (V, A)$ be a complete directed graph with cost function $c: A \rightarrow Z^+$, and $G_0 = (V, A')$, a subgraph of G. Let C^* be the cost of an optimal augmentation A^* that makes G_0 strongly connected, and let $\hat{C}$ be the cost of the edges A'' generated by algorithm STC. Then

$\hat{C}/C^* \leq 2.$

Furthermore, there exists a family of examples for which the bound can be approached arbitrarily closely.

For the proof, see [5].

4. Bridge Connectivity Augmentation Algorithm

We first consider the problem of bridge connecting a graph G_0, where G_0 is connected. In Section 2, we have shown that this problem is NP-hard. We make several observations about solution strategies, and then proceed to a description of our algorithm. First, we note that we may actually consider bridge connecting a subgraph that is a tree. If we are given G_0 with nontrivial bridge-connected components, we shrink the vertex sets of the bridge-connected components into corresponding single vertices, resulting in a tree whose edges are bridges in G_0. The second observation is that an algorithm should try to connect two leaves together at a time. If we use a greedy heuristic of joining a leaf to the nearest vertex, the edge set so identified may be arbitrarily more expensive than the optimal augmenting set. Our final observation is that there is an element of directedness to this problem, that at first glance appears undirected.

Our algorithm is as follows. We arbitrarily choose a leaf as a root r in our tree, and direct all edges in the tree toward r. We then use a minimum weight arborescence algorithm to find a directed tree out of r. The edges in the arborescence, when added to the directed tree, will form a strongly connected graph. The corresponding edges will form a bridge-connected graph.

Because we are transforming an undirected problem into a directed problem, we develop the following distance function, which counteracts poor choices in directing edges in the tree. Let $d(u, v) = \min\{c(x, y) \mid u$ and v are on a path from x to y in $T\}$. This distance function can be computed in $O(|V|^2)$ time by algorithm DIST [5].

If G_0 is not a tree (but is of course connected), we first shrink all vertices in a bridge-connected component to a single vertex representing that bridge-connected component. Thus G is shrunk to a complete graph $G_s = (V_s, E_s')$. The cost c' of an edge (u, v) is the minimum of $\{\infty\} \cup \{c(x, y) \mid (x, y) \varepsilon E - E'$ and x is in the component represented by u, and y is in the component represented by $v\}$. The edge (x, y) so used is referenced by a backpointer $b(u, v) = (x, y)$.

We now proceed to the specification of algorithm BRC, which bridge-connects a connected subgraph. We note that our handling of distance d', in step 4b, allows us to slide back on edges in the tree, and also forces us to use node r as the root of the arborescence. Also note that we of course eliminate duplicates that might occur from insertions into E''.

Algorithm BRC

```
Input: $G = (V, E)$, a complete undirected graph with weight function
  $c: E \to Z^+$, and $G_0 = (V, E')$, a spanning subgraph of $G$.
Output: A set of edges $E'' \subset E - E'$ such that $(V, E' \cup E'')$ is bridge-connected.
1.   Shrink $G$ to $G_s = (V_s, E_s)$ and $G_0$ to $T = (V_s, E'_s)$, such that each
     bridge connected component in $G_0$ is mapped to a different vertex in
     $V_s$. Find weight function $c': E_s \to Z^+ \cup \{\infty\}$ and $b: E_s \to E$.
2.   Find $d: E_s \to Z^+$ and $b': E_s \to E_s$ using algorithm DIST.
3.   Choose any leaf in $V_s$ to be $r$. Let $A'$ be the set of directed
     edges generated by directing each edge in $E'_s$ toward $r$. Let
     $T' = (V_s, A')$.
4.a. Let $A = \emptyset$. For each edge $(u, v) \in E_s$, insert $\langle u, v\rangle$ and $\langle v, u\rangle$ into
     $A$. Set $d'\langle u, v\rangle = d'\langle v, u\rangle = d(u, v)$.
  b. If $\langle u, v\rangle \in A'$, then set $d'\langle u, v\rangle = 0$. For all $u$, set $d'\langle u, r\rangle = \infty$.
5.   Find a minimum weight arborescence $T'' = (V_s, A'')$ on $(V_s, A)$, using $d'$.
6.   For each edge $\langle u, v\rangle$ in $A''$ with $d'\langle u, v\rangle > 0$, insert the corre-
     sponding edge $b(b'(u, v))$ into $E''$. □
```

Theorem 3. Algorithm BRC generates a set E'' such that $(V, E' \cup E'')$ is bridge-connected.

The proof will be omitted here. For more details, see [5].

Lemma 2. Let $E^* \subset E - E'$ be a minimum cost set of edges such that $(V, E' \cup E^*)$ is bridge-connected. Then there is an augmentation $A''' \subset A - A'$, of cost no more than twice that of E^*, such that $(V_s, A' \cup A''')$ is strongly connected.

Proof. Let (u, v) be an edge in E^*. Let (x, y) be the corresponding edge when V is shrunk to V_s. Consider the weakly directed path between x and y in T'. If the edge directions do not change along the path, direct (x, y) to complete a directed cycle and insert it into A'''. If the edge directions change, then they change at most once, say at vertex w. Insert $\langle x, y\rangle$ and $\langle w, x\rangle$ into A'''. By the definition of the distance functions d', d and c', $d'\langle w, x\rangle \leq c'(x, y) \leq c(u, v)$. Hence each edge (u, v) in E^* accounts for at most two directed edges in A''', each of cost no more than (u, v). Since every edge in T must be on a cycle containing exactly one edge in E^*, then every directed edge in T' must be on a directed cycle in $(V_s, A' \cup A''')$, and hence this graph is strongly connected. □

Theorem 4. Let $G = (V, E)$ be a complete undirected graph with cost function $c: E \to Z^+$ and $G_0 = (V, E')$ a connected subgraph of G. Let C^* be the cost of an optimal augmentation E^* that makes G_0 bridge-connected, and let $\hat{C}$ be the cost of the edges E'' generated by algorithm

BRC. Then

$\hat{C}/C^* \leq 2.$

Furthermore, there exists a family of examples with all edges of cost 1 such that the bound can be approached arbitrarily closely.

Proof. From Lemma 2, we know that there is an augmentation A''' of cost no greater than $2C^*$, such that $(V_s, A' \cup A''')$ is strongly connected. From A''', we may construct an arborescence for (V, A') using d', and with any particular vertex that we choose to be the root. The cost $\hat{C}$ will be no greater than the cost of the minimum-cost arborescence, which will be no greater than the cost of the induced arborescence, which will be no greater than $2C^*$.

To see that the bound is approachable, consider $G_0 = (V, E')$ which is a binary tree with n leaves. Let $c(u, v) = 1$ for all $u, v \varepsilon V$, $u \neq v$. The algorithm will choose a leaf at random to be the root r, and direct all tree edges toward r. A spanning arborescence must have a directed edge entering each of the remaining $n - 1$ leaves, so that a potential solution is the set of edges connecting r with each other leaf, which is of cost $\hat{C} = n - 1$. An optimal solution will have $n/2$ edges, each edge connecting two leaves, for a cost $C^* = n/2$. □

If the subgraph G_0 is not connected, then we modify BRC so that the connected components are joined in minimum cost fashion. Prim's algorithm for finding a minimum spanning tree [13] may be used to accomplish this in $O(|V|^2)$ time. Since the cost of the minimum connection will be no larger than the cost of an optimum bridge-connection C^*, the worst-case bound on this algorithm is $\hat{C}/C^* \leq 3$. If the cost function satisfies the triangle inequality, then we can do much better with another algorithm [6].

5. Biconnectivity Augmentation Algorithm

Finding an augmentation that biconnects a graph seems to involve difficulties that are not present in the bridge connectivity augmentation problem. In the case of finding an optimal augmentation for unweighted graphs, the $O(|V| + |E|)$ algorithm for biconnection in Rosenthal and Goldner [14] is considerably more involved than the $O(|V| + |E|)$ algorithm for bridge connection in Eswaran and Tarjan [4]. We have found a similar increase in complication in the weighted case in moving from bridge-connectivity to biconnectivity augmentation approximation algorithms.

The block-cutvertex tree $T_s = (V_s, E_s')$ is used to handle not only biconnected components (blocks), but cutvertices as well. Let $G^i = (V^i, E^i)$, $1 \leq i \leq t$, be the blocks of G_0 and let V_c be the cutvertices of G_0, and let V_b be a set of new vertices, called block vertices, representing the blocks of G_0. Let $V_s = V_c \cup V_b$. Let E_s' be such that each edge connects a block vertex with a cutvertex, such that the cutvertex is in the corresponding block. Hence cutvertices and block vertices alternate in the block-cutvertex tree T_s. The block-cutvertex graph $G_s = (V_s, E_s)$ induced by G_0 will have $E_s = \{(v_i, v_j) | v_i, v_j \varepsilon V_s \wedge i \neq j\}$.

We define cost function c' on E_s in terms of cost function c, but being careful about the way cutvertices are handled. If $v_i, v_j \varepsilon V_c$, then set $c'(v_i, v_j) = c(v_i, v_j)$. If $v_i \varepsilon V_b$ and $v_j \varepsilon V_c$, then set $c'(v_i, v_j) = \min\{\{\infty\} \cup \{c(v_k, v_j) | v_k \varepsilon (V^i - V_c)\}\}$. If $v_i, v_j \varepsilon V_b$, then set $c'(v_i, v_j) = \min\{\{\infty\} \cup \{c(v_k, v_m) | v_k \varepsilon (V^i - V_c) \wedge v_m \varepsilon (V^j - V_c)\}\}$. Our cost function c' has been defined as though we do not include cutvertices in blocks. We can do this since we have individual vertices in V_s representing the cutvertices, and hence need not include them in the block vertices. We allow the possibility of an edge with cost ∞, since all vertices in some block may be cutvertices. Finally, for each edge (u, v) in E_s, let $b(u, v)$ be a pointer to an edge (x, y) in E with $c(x, y) = c'(u, v) < \infty$.

Before presenting the algorithm, we give an example, in Figure 1, of how the cutvertices must be handled carefully. In Figure 1a, we have a block-cutvertex tree. In Figure 1b, we give a directed tree without introducing the new vertices of step 3b. The dotted edges give a strong connection for the directed tree, but a cutvertex still remains in the corresponding undirected graph. In Figure 1c, we show a directed tree with the new vertices included. If the same augmentation is used, as shown, then this augmentation does not strongly connect the tree. We cannot use an edge $\langle c_1, b_4 \rangle$ in the augmentation, since no such edge is introduced in step 1a. Hence an augmentation such as that in Figure 1d must be used, and this both strongly connects the tree, and also induces a biconnection in the original graph.

Algorithm BIC

Input: $G = (V, E)$, a complete undirected graph with weight function $c: E \to Z^+$, and $G_0 = (V, E')$, a spanning subgraph of G.

Output: A set of edges $E'' \subset E - E'$ such that $(V, E' \cup E'')$ is biconnected.

1. Shrink G_0 to its block-cutvertex tree $T_s = (V_s, E'_s)$ and G to $G_s = (V_s, E_s)$, the block-cutvertex graph induced by G_0. Find weight function $c': E_s \to Z^+ \cup \{\infty\}$ and $b: E_s \to E$.

2. Find $d: E_s \to Z^+$ and $b': E_s \to E_s$ using algorithm DIST.

3.a. Choose any leaf in V_s to be r. Let A' be the set of directed edges generated by directing each edge in E'_s toward r. Set V'_s to V_s.

3.b. For each edge $\langle u, v \rangle$ in A' such that $v \varepsilon V_c$, insert new vertex x_{uv} in V'_s, and replace $\langle u, v \rangle$ in A' with $\langle u, x_{uv} \rangle$ and $\langle x_{uv}, v \rangle$. Let $T'_s = (V'_s, A')$.

4.a. Set A to A'. For each edge $(u, v) \varepsilon E_s - E'_s$, do the following. If $u \notin V_c$ or there is no path from v to u in T'_s, then insert $\langle u, v \rangle$ into A with $d'\langle u, v \rangle = d(u, v)$. Otherwise, let x be the (new) vertex preceding u in T'_s, and insert $\langle x, v \rangle$ into A with $d'\langle x, v \rangle = d(u, v)$ and set $b'(x, v) = b'(u, v)$. Perform the above with the roles of u and v reversed.

4.b. If $\langle u, v \rangle \varepsilon A'$, then set $d'\langle u, v \rangle = 0$. For all u, set $d'\langle u, r \rangle = \infty$.

5. Find a minimum cost arborescence $T''_s = (V'_s, A'')$ on (V'_s, A') using d'.

6. For each edge $\langle u, v\rangle$ in A'' with $d'\langle u, v\rangle > 0$, insert the corresponding edge $b(b'(u, v))$ into E''. ☐

Theorem 5. Algorithm BIC generates a set E'' such that $(V, E' \cup E'')$ is biconnected.

For the proof, see [5].

Lemma 3. Let $E^* \subset E - E'$ be a minimum cost set of edges such that $(V, E' \cup E^*)$ is biconnected. Then there is an augmentation $A''' \subset A - A'$ of cost no more than twice that of E^*, such that $(V'_s, A' \cup A''')$ is strongly connected.

Proof. We describe an algorithm for generating A''' from E^*. Start with $A''' = \emptyset$. Mark all vertices inaccessible except the root of T'_s. While not all vertices are accessible, do the following: Choose an accessible vertex $w \varepsilon V_b$ such that w is on the path from x to y in T_s for some edge (x, y) corresponding to an unexamined edge $(u, v) \varepsilon E^*$. If $w \neq y$, insert $\langle w, y\rangle$ into A''', and if $w \neq x$, insert $\langle w, x\rangle$ into A'''. Mark all vertices on the weakly directed path from x to y in T'_s accessible.

Since $w \varepsilon V_b$ edges $\langle w, y\rangle$ and $\langle w, x\rangle$ exist in A and have d' cost no greater than $c(u, v)$. Also, if w is accessible from the root then introduction of the two directed edges will make all vertices on the weakly directed path from x to y accessible from the root.

It remains to show that an edge (u, v) from E^* and suitable accessible vertex w always exist. At any point in the execution of the above algorithm the vertices V_e marked as accessible represent a set of vertices that is biconnected. If E^* has not been completely examined then there must be an edge (x, y) corresponding to an unexamined edge (u, v) in E^* such that the path between x and y in T_s contains more vertices in V_e than just one cutvertex. Otherwise E^* cannot neutralize all cutvertices from V_c. If there is more than one vertex from V_e on a path between x and y in T_s, then there must be a $w \varepsilon V_b \cap V_e$ on this path, since blockvertices and cutvertices alternate in T_s. ☐

Theorem 6. Let $G = (V, E)$ be a complete undirected graph with cost function $c: E \to Z^+$ and $G_0 = (V, E')$ a connected subgraph of G. Let C^* be the cost of an optimal augmentation E^* that makes G_0 biconnected, and let $\hat{C}$ be the cost of the edges E'' generated by algorithm BIC. Then

$$\hat{C}/C^* \leq 2.$$

Furthermore, there exists a family of examples with all edges of cost 1 such that the bound can be approached arbitrarily closely.

Proof. Similar to the proof of Theorem 4. ☐

Like algorithm BRC, algorithm BIC has the complexity $O(|V|^2)$. If the subgraph G_0 is not connected, then we first connect the graph, in minimum cost fashion, and then run BIC. The worst-case bound is 3, as in the case of bridge connection.

[1] Y. J. Chu and T. H. Liu, "On the shortest arborescence of a directed graph," Sci. Sinica 14 (1965) 1396-1400.

[2] S. Cook, "The complexity of theorem-proving procedures," Proc. 3rd Ann. ACM Sympo. on the Theory of Computing (1971) 151-158.

[3] J. Edmonds, "Optimum branchings," J. Res. Nat. Bur. Standards Sect. B 71 (1967) 233-240.

[4] K. P. Eswaran and R. E. Tarjan, "Augmentation problems," SIAM J. Comput. 5 (1976) 653-665.

[5] G. N. Frederickson and J. Ja'Ja', "Approximation Algorithms for Several Graph Augmentation Problems," Technical Report CS-79-22, The Pennsylvania State University, University Park, PA 16802.

[6] G. N. Frederickson and J. Ja'Ja', "On the relationship between the biconnectivity augmentation and traveling salesman problems," Technical Report CS-79-41, The Pennsylvania State University, University Park, PA 16802.

[7] M. R. Garey and D. S. Johnson, "Approximation algorithms for combinatorial problems: an annotated bibiliography," in Algorithms and Complexity: Recent Results and New Directions, J. F. Traub, ed., Academic Press, New York (1976) 41-52.

[8] M. R. Garey and D. S. Johnson, Computers and Intractibility: A Guide to the Theory of NP-completeness, W. H. Freeman, San Francisco (1979).

[9] F. Harary, Graph Theory, Addison-Wesley, Reading, Mass. (1969).

[10] D. S. Johnson, "Approximation algorithms for combinatorial problems," JCSS 9 (1974) 256-278.

[11] R. M. Karp, "Reducibility among combinatorial problems," in Complexity of Computer Computations, R. E. Miller and J. W. Thatcher, eds., Plenum Press, New York (1972) 85-104.

[12] R. M. Karp, "On the complexity of combinatorial problems," Networks 5 (1975) 45-68.

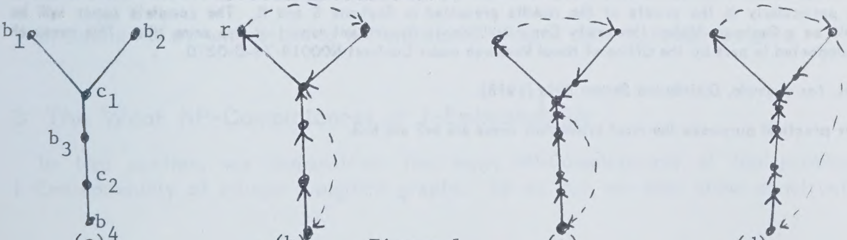
[13] R. C. Prim, "Shortest connection networks and some generalizations," Bell Sys. Tech. J. 36 (1957) 1389-1401.

[14] A. Rosenthal and A. Goldner, "Smallest augmentations to biconnect a graph," SIAM J. Comput. 6 (1977) 55-66.

[15] R. Tarjan, "Finding optimum branchings," Networks 7 (1977) 25-35.

[16] R. Tarjan, private communication (1979).

[17] A. C. Yao, "An $O(|E|\log\log|V|)$ algorithm for finding minimum spanning trees," Info. Proc. Letters 4 (1975) 21-23.



(a) (b) Figure 1 (c) (d)

Embeddability of Weighted Graphs in k-Space is Strongly NP-Hard

(Extended Summary)[1]

JAMES B. SAXE
Computer Science Department
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

Abstract--In this paper we investigate the complexity of embedding edge-weighted graphs into Euclidean spaces: Given an (incomplete) edge-weighted graph, G, can the vertices of G be mapped to points in Euclidean k-space in such a way that any two vertices connected by an edge are mapped to points whose distance is equal to the weight of the edge? We prove that the preceding problem is NP-Hard (by reduction from 3-Satisfiability), even when k=1 and the edge weights are restricted to take on the values 1 and 2. Related results are shown for the problem of testing the uniqueness of a known embedding and for variations involving inexact edge weights.

1. Introduction

In many applications of distributed sensor networks[2] there arises the problem of determining the locations of sensors from incomplete (and possibly errorful) information about their distances from each other and from fixed landmarks. This prompts us to ask the following geometric questions:

> -Given an incompletely specified distance matrix for a set of points in k-space,[3] when is the complete distance matrix uniquely determined?
>
> -Assuming the disiance matrix to be uniquely determined, what is the computational complexity of actually finding the unspecified distances?

In this paper we consider the closely related problem of embeddability:

> -Given a (purported) incompletely specified distance matrix for a set of points in k-space, determine whether there can actually exist a set of points satisfying that matrix.

In Section 2 we introduce definitions that will allow us to phrase several forms of the embeddability problem in terms of edge-weighted graphs. In Section 3, we give a simple proof that a 1-dimensional version of the embeddability problem is NP-Complete. In Section 4, we show the more difficult and surprising result that this same 1-dimensional problem is strongly NP-Complete in the sense of Garey and Johnson [1979] and extend this result to higher dimensions. In Section 5 we address some naturally arising questions concerning the suitability of the Turing Machine model for a problem that inherently involves real numbers, and show that the proofs used in Section 4 have relevance to an "approximate embeddability" problem on the reals. In Section 6 we discuss versions of the problem in which one way to complete an incompletely specified distance matrix is known and it is desired to determine whether a second solution exists. We show that these versions are no easier than corresponding versions studied earlier in the paper. Finally, the contributions of the paper are summarized in Section 7.

---

[1] In order to meet the space constraints of these *Proceedings*, a number of details have been omitted from this paper, particularly in the proofs of the results presented in Sections 5 and 6. The complete paper will be available as a Carnegie-Mellon University Computer Science Department report of this same title. This research was supported in part by the Office of Naval Research under Contract N00014-76-C-0370.

[2] See, for example, *Distributed Sensor Nets* [1978].

[3] For practical purposes the most interesting cases are k=2 and k=3.

2. Fundamental Concepts

We begin by introducing the concepts of weighted graph and embedding:

Definitions:

A weighted graph, G = <V,E,W>, is an ordered triple such that each element of E is an unordered pair of distinct elements of V and W is a function mapping E into $[0,\infty)$. The elements of V are called the vertices of G. The elements of E are called the edges of G. For each edge, e, of G, the real number W(e) is called the weight of e in G (or simply the weight of e).

Definitions:

Let G = <V,E,W> be a weighted graph, and let k be a positive integer. Then an embedding of G in k-space is a function, f, mapping V into the k-dimensional Euclidean space, $\mathbb{R}^k$, such that, for each edge, e = {v,w}, of G, $|f(v)-f(w)| = W(e)$. G is said to be embeddable in k-space, or k-embeddable, iff there exists an embedding of G in k-space.

For any positive integer, k, the problem of k-embeddability may now be stated as follows:

Problem (k-Embeddability):

Given an arbitrary weighted graph, G, determine whether G is k-embeddable.

In Sections 3 and 4 we will wish to restrict the class of weighted graphs under consideration, so that the notion of NP-Completeness (which is defined in terms of Turing machines) will make sense in relation to Embeddability. We therefore introduce the following definition.

Definition:

Let S be any subset of $[0,\infty)$. Then, an S-weighted graph is a weighted graph, G, such that the weight of each edge of G is an element of S. We will generally refer to $\mathbb{Z}^+$-weighted graphs as integer-weighted graphs.

In Section 5 we will return to the question of graphs with real edge weights.

3. The Weak NP-Completeness of 1-Embeddability

In this section, we demonstrate the weak NP-Completeness of the problem of 1-Embeddability of integer-weighted graphs. To do this, we first show constructively that 1-Embeddability is in NP. We then use a reduction from Partition[4] to show completeness.

Theorem 3.1:

1-Embeddability of integer-weighted graphs is in NP.

Proof:

To check the 1-embeddability of any integer-weighted graph, a NDTM need only

1. Partition the graph into disjoint connected subgraphs,
2. Guess the direction of each edge of the graph, and
3. Check the consistency of each disjoint connected subgraph.

These operations can clearly be carried out in (nondederministic) polynomial time. □

Theorem 3.2:

1-Embeddability of integer-weighted graphs is NP-Complete.

Proof:[5]

We will show the NP-Completeness of 1-Embeddability by reduction from Partition. Let $S = \{a_1, a_2, \ldots, a_n\}$ be a multiset of positive integers. In polynomial time we may construct from S a description of the cyclic graph $G = <V,E,W>$ whose edge weights are the a_i, that is

$$V = \{v_0, \ldots, v_{n-1}\},$$
$$E = \{\{v_i, v_{(i+1 \bmod n)}\} \mid 0 \le i < n\}, \text{ and}$$
$$W = \{(\{v_i, v_{(i+1 \bmod n)}\}, a_i) \mid 0 \le i < n\}.$$

If f is an embedding of G in the line, then the multisets

$$S_1 = \{a_i \mid f(v_i) < f(v_{(i+1 \bmod n)})\} \text{ and}$$
$$S_2 = \{a_i \mid f(v_i) > f(v_{(i+1 \bmod n)})\}$$

constitute a partition of S into two pieces whose sums are equal. Similarly, any such partition of S yields a 1-Embedding of G. □

---

[4] The Partition problem calls for partitioning a (multi-)set of integers into two subsets with equal sums, and is known to be NP-Complete; see Garey and Johnson [1979].

[5] The construction used in this theorem and that used in the proof of Lemma 4.4 were independenty developed by Yemini [1978], who used them to show the (weak) NP-Completeness of 2-Embeddability of integer-weighted graphs.

4. The Strong NP-Completeness of 1-Embeddability

We now come to our key theorem, which asserts that the problem of determining whether an integer-weighted graph is embeddable in the line remains NP-Complete even if the edge weights are restricted to be no greater than four.

Theorem 4.1
1-Embeddability of $\{1,2,3,4\}$-weighted graphs is NP-Complete.

Proof:[6]
Our proof consists of a reduction from 3-Satisfiability (which was shown to be NP-Complete by Cook [1971]) to 1-Embeddability of $\{1,2,3,4\}$-weighted graphs. Let E be any Boolean expression in conjunctive normal form with three literals in each clause. Our goal will be to construct a $\{1,2,3,4\}$-weighted graph, G, which is embeddable iff E is satisfiable. We let n be the number of variables occurring in E and m be the number of clauses in E. Throughout this proof, we will use the convention that the variables of E will be indexed by "i" (which will therefore range from 1 through n), the clauses of E will be indexed by "j" (ranging from 1 through m), and the literals within each clause will be indexed by "k" (ranging from 1 through 3). Thus E has the form

$$E = \prod_{1 \le j \le m} C_j,$$

where each clause, C_j, has the form

$$C_j = \sum_{1 \le k \le 3} L_{j,k},$$

and each literal, $L_{j,k}$, has the form

$$L_{j,k} = X_i \quad \text{or} \quad L_{j,k} = \bar{X}_i$$

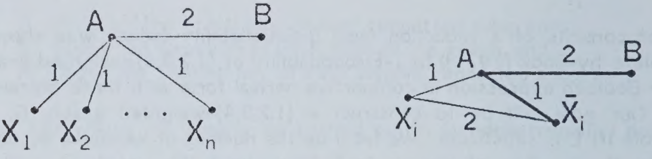
for some i, $1 \le i \le n$. We will also use throughout the proof the convention that "f" represents a hypothetical 1-embedding of G (or of the part of G we have constructed so far).

To construct G, we will use the "building blocks" shown in Figure 4.1. We begin with the subgraph shown in Figure 4.1(a). We assume without loss of generality that $f(A) = 0$ and $f(B) = 2$. This assumption constrains f to assign each of the X_i (which we identify with the variables of E) to 1 or -1 (which we identify with the Boolean values TRUE and FALSE, respectively). Note that each possible mapping of the X_i into $\{1,-1\}$ corresponds to some assignment of truth values to the X_i. In the remaining steps of the construction, we will add edges which have precisely the effect of constraining f to map the X_i to $\{1,-1\}$ in such a way that the corresponding assignment of the X_i satisfies E.

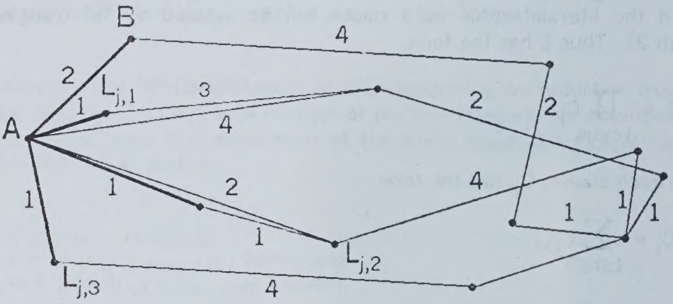
---

[6] Another NP-Complete problem involving a form of graph embedding is the Bandwidth Minimization Problem (see Papadimitriou [1976]). In the full paper we will also exhibit a reduction from Bandwidth Minimization to Embeddability. This reduction will suffice to show the strong NP-Completeness of 1-Embeddability of integer-weighted Graphs, although it is somewhat less economical than the construction given here.

The next step in our construction is to augment G by adding the edges shown in Figure 4.1(b) for each i, $1 \le i \le n$. The heavy lines in that figure represent already-existing edges. We now have vertices $\bar{X}_i$ such that for each variable, X_i, f maps X_i to 1 (TRUE) iff it maps $\bar{X}_i$ to -1 (FALSE), and vice-versa. The possible mappings from $\{X_i\} \cup \{\bar{X}_i\}$ to $\{1,-1\}$ under f now correspond precisely to the possible (consistent) truth assignments of the X_i and $\bar{X}_i$, but still without regard to whether those assignments satisfy E.



(a) Implementation of variables. (b) Implementation of a negative literal.



(c) Implementation of a disjunctive clause.

Figure 4.1 Building blocks for transforming an expression in 3-DNF to a graph.

For the final step of our construction, we add the edges indicated in Figure 4.1(c) for each j, $1 \le j \le m$. The vertices $L_{j,k}$ are identified with the X_i and $\bar{X}_i$ precisely as the corresponding literals, $L_{j,k}$, are formally identical with the X_i and $\bar{X}_i$. Once again, the heavy lines indicate edges which were present at earlier stages of the construction. Careful study of the graph in Figure 4.1(c) will reveal that it is impossible to embed it in the line in such a way that A is sent to 0, B is sent to 2, and all three of the $L_{j,k}$ are sent to -1 (FALSE), but if one or more of the $L_{j,k}$ are to be sent to 1 (TRUE), then an embedding is possible (in fact, exactly one such embedding is possible). Thus, for each j, $1 \le j \le m$, the effect of the edges in Figure 4.1(c) is precisely to constrain f to map the X_i to $\{1,-1\}$ in such a way that the corresponding truth assignment for the X_i satisfies clause C_j.

The effect of all the edges of G is therefore to constrain f to map the X_i to $\{1,-1\}$ in such a way that the corresponding assignment of truth values to the X_i satisfies E. If there is no such assignment then G is not 1-embeddable. If there are any assignments satisfying E, then for each such assignment G can be (uniquely)

1-embedded by a function sending A to 0 and B to 2 and mapping the X_i to $\{1,-1\}$ in accordance with that assignment. Finally, it is clear that the preceding construction can be carried out in polynomial time. This completes the proof. ☐

For future reference, we note that the construction used in the preceding proof is such that the 1-embeddings of G are in one-to-one correspondence (up to translation and reflection) with the truth assignments that satisfy E. We note also that the preceding theorem immediately yields the following result:

Corollary 4.2:
1-Embeddability of integer-weighted graphs is strongly NP-Complete.

Proof:
It suffices to note that translation of a sequence of numbers in $\{1,2,3,4\}$ from binary to unary can be accomplished in linear time and causes only a constant factor increase in the length of the input. ☐

We may also immediately derive:

Corollary 4.3:
1-Embeddability of $\{1,2\}$-weighted graphs is NP-Complete.

Proof:
Consider the graphs shown in Figure 4.2. By replacing single edges of weights 3 and 4 with configurations T_3 and T_4, respectively, we can reduce any $\{1,2,3,4\}$-weighted graph, G, to a $\{1,2\}$-weighted graph, H, that is 1-embeddable iff G is 1-embeddable. ☐

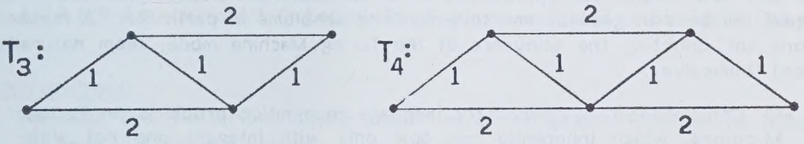


Figure 4.2. Building long "edges" from short edges.

In fact, for any positive integer, k, the graph H so constructed will be k-embeddable iff G is k-embeddable. We may use this fact to prove our next lemma.

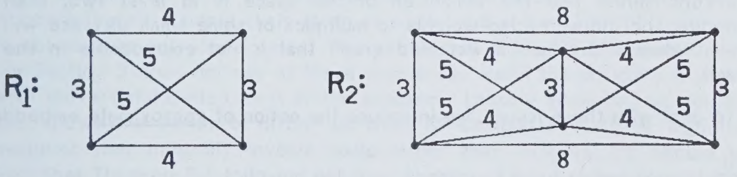


Figure 4.3. Gadgets for adding a dimension.

Lemma 4.4:
For every positive integer, k, k-Embeddability of $\{1,2\}$-weighted graphs is NP-Hard.

Proof:

Consider the graphs shown in Figure 4.3. Given any {1,2}-weighted graph, G, each edge of G having weight 1 may be replaced by the R_1 and each edge of weight 2 by R_2, yeilding a graph, H, which, for any positive integer, k, is embeddable in (k+1)-space iff G is embeddable in k-space. By the methods of Theorem 4.3, H may be transformed into a {1,2}-weighted graph, J, that is embeddable in precisely those spaces in which H is embeddable. The transformation from G to J involves only a constant factor increase in the length of a specification of the graph and can clearly be accomplished in polynomial time. It follows by mathematical induction that, for any positive integer, k, 1-Embeddability is polynomial-time reducible to k-Embeddability for {1,2}-weighted graphs. ☐

Once again, we note that the (k+1)-embeddings of J will be in one-to-one correspondence (up to rotation and reflection) with the k-embeddings of G. Finally, Theorem 4.4 gives us the following result.

Corollary 4.5:

Let k be any positive integer. Then k-Embeddability of integer-weighted graphs is strongly NP-Hard.

Proof:

This result follows from Lemma 4.4 and the same reasoning used in the proof of Corollary 4.2. ☐

5. Graphs with Real-Valued Edge Weights

We will now discuss the applicability of NP-Completeness to problems whose inputs are real numbers in general, and to embedding problems in particular. A number of reasons for doubting the relevance of the Turing Machine model seem naturally to present themselves.

-NP-Completeness is defined for language recognition problems on Turing Machines, which inherently can deal only with integers and not with arbitrary reals.

-Given a "random" embedding of an unweighted graph into a Euclidean space, any two of the edge weights induced by the embedding will be incommensurable with probability 1. Moreover, if the graph is overconstrained and the dimension of the space is at least two, then rounding the induced edge-weights to multiples of some small distance will almost always produce a weighted graph that is not embeddable in the space.

In order to deal with these issues, we introduce the notion of approximate embeddings.

Definitions.

Let G be a weighted graph and ϵ be a positive real number. Then an ϵ-approximate k-embedding of G is a function, f, that maps the vertices of G into Euclidean k-space such that for every edge, $\{u,v\}$, of G, $1-\epsilon < |f(u)-f(v)|/W(\{u,v\}) < 1+\epsilon$. If such an embedding exists, then G is said to be ϵ-approximately k-embeddable.

Given a positive integer, k, and two reals, ϵ_1 and ϵ_2, such that $0 < \epsilon_1 < \epsilon_2$, we may now define the following more "robust" embeddability problem:

Problem (ϵ_1,ϵ_2-Approximate k-Embeddability):

Given a weighted graph, G, assert correctly either (1) that G is ϵ_2-approximately k-embeddable (this is called accepting G) or (2) that G is not ϵ_1-approximately k-embeddable (this is called rejecting G).

Note that if the least ϵ for which G is ϵ-approximately k-Embeddable lies in the interval $[\epsilon_1,\epsilon_2]$, then it is permissible either to accept or to reject G. In this problem definition, we have attempted to capture, without introducing inordinately many complexities of detail, the essential problem of embedding as it would apply to real computers given inexact data.

We now wish to investigate the computational complexity of ϵ_1,ϵ_2-Approximate Embeddability problems. Is it possible, for example, to solve all such problems in time polynomial in the size of a specification of G (where the degree of the polynomial, or even just the "constant" factor, depends on $(\epsilon_2-\epsilon_1)^{-1}$)?

It turns out that such polynomial solutions are not possible in the general case (assuming that $P \neq NP$). In particular, we have the following result.

Theorem 5.1:

1/18,1/9-Approximate 1-Embeddability of integer-weighted graphs is NP-Complete.

Sketch of Proof:

We note that the embeddability properties of the graphs used in the proof of Theorem 4.1 depend only on cycles of length no greater than 16 having edges whose lengths are multiples of 1. It follows from this that for such graphs ϵ-Approximate 1-Embeddability is equivalent to ordinary 1-embedability for any $\epsilon <$ 1/8, and the desired result is at hand. ☐

It is interesting to note that Approximate 1-Embeddability problems restricted to graphs consisting of a single cycle (such as were used in the proof of Theorem 3.2) are always solvable in polynomial time.[7] This shows that the weak NP-Completeness result given in Section 3 does not say all there was to say about the difficulty of the practical (*i.e.*, with inexact data, etc.) form of the problem. Loosely speaking, we could say that we have shown the notion of strong *vs.* weak NP-Completeness to be significant even for problems that naturally involve reals rather than integers. It should be noted, however, that Theorem 5.1 followed not from Theorem 4.4 but rather from the particular construction used in the proof of Theorem 4.4.

---

[7] This follows from the existence of fast approximation algorithms for Partition. See, for example, Lawler [1977].

6. Ambiguous Embedding Problems

Another variation on the embeddability problem that may arise in practical applications is that of "ambiguity of solution." Given an incomplete weighted graph and some embedding of that graph into a Euclidean space, we may wish to know whether the given into a Euclidean space, we may wish to know whether the given embedding is unique.[8] To pose the problem more precisely, we introduce the following definitions.

Definitions:

Let G be a weighted graph and k be a positive integer. Then two k-embeddings, f and g, of G are said to be congruent iff for each two vertices, u and v, of G, $|f(u)-f(v)| = |g(u)-g(v)|$. A k-embedding, f, of G is said to be unique (up to congruence) iff every k-embedding of G is congruent to f, and in this case G is said to be uniquely k-embeddable. If G has two or more non-congruent k-embeddings, then G is ambiguously k-embeddable.

For any positive integer, k, we may now define the problem of Ambiguous k-Embedding as follows:

Problem (Ambiguous k-Embedding):

Given a weighted graph, G, and a k-embedding, f, of G, determine whether G is ambiguously k-embeddable (*i.e.*, whether there exists a k-embedding of G which is not congruent to f).

In the full paper, we will show that Ambiguous k-Embeddability is (strongly) NP-Hard for all positive k and that it is NP-Complete for k=1. Here we have space only for a brief overview of the steps of the proof, which go as follows:

1. We define Ambiguous 3-Satisfiability and Ambiguous 4-Satisfiability in a manner analogous to the definition of Ambiguous k-Embedding.
2. We show the NP-Completeness of Ambiguous 4-Satisfiability by reduction from 3-Satisfiability.
3. We show the NP-Completeness of Ambiguous 3-Satisfiability by reduction from Ambiguous 4-Satisfiability.
4. We show the (strong) NP-Completeness of Ambiguous 1-Embedding by reductiion for Ambiguous 3-Satisfiability, using the fact that the construction used in the proof of Theorem 4.1 "preserves uniqueness" (up to congruence).
5. We show the (strong) NP-Hardness of Ambiguous k-Embedding for k>1 by reduction from Ambiguous (k-1)-Embedding.

7. Conclusions

The results of this paper fall into two classes, those of interest to persons concerned with embedding problems (such as the sensor positioning problem) and those that are of more general theoretical interest. To those concerned with finding efficient solutions to the Embedding problem (given a weighted graph, find "the" embedding), these results say what all NP-Completeness results say: "You are trying to solve the wrong problem." Rather than looking for an efficient worst-case algorithm, it would be more promising to seek an algorithm that gives good performance in cases which arise in practice (for example, cases in which the graph is highly overconstrained). Pursuing this topic, in the full paper we will discuss a linear-time algorithm due to Shamos [1978] for determining whether a given *complete* graph is k-embeddable (for any fixed k). His algorithm assumes a model in which real arithmetic operations can be performed in constant time.

---

[8] For example, are the nodes of our sensor network really where we think they are, or might they be in some very different configuration?

The most specific result of theoretical interest is our discovery of some new ***strongly*** NP-Hard geometric problems, and our use of some interesting gadgets to carry out the proofs of NP-Hardness. Of more general interest are the two new classes of problems introduced in Sections 5 and 6. The "ϵ_1,ϵ_2-approximate" problems introduced in Section 5 offer a new way of looking at the notion of NP-Completeness in the context of problems involving continuous variables. As we have seen, weak NP-Completeness may not say all there is to say in this context. "Ambiguous solution" problems address the question of determining whether a known solution to a problem is in fact the unique solution. In Section 6, we exhibited a fundamental NP-Complete problem, 3-Satisfiabilty, whose ambiguous version is also NP-Complete, and exhibited a method for obtaining new NP-Completeness results for such ambiguous solution problems, namely the use of reductions that preserve uniquenes of solution.

Acknowledgements

The author would like to thank Steven E. Saunders for leading him to study the problem of Embeddability, Michael I. Shamos for helpful discussions, and Jon L. Bentley for his encouragement and advice during the preparation of this document.

References

[1] Cook, S. A. "The complexity of theorem proving procedures." *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing.* Association for Computing Machinery, New York (1971). pp. 151-158.

[2] *Distributed Sensor Nets.* Proceedings of a conference sponsored by the Information Processing Techniques Office, Defense Advanced Research Projects Agency and hosted by Carnegie-Mellon University, December, 1978.

[3] Garey, M. R. and D. S. Johnson. *Computers, Complexity, and Intractability.* Freeman, San Francisco (1979).

[4] Lawler, E. L. "Fast approximation algortihms for knapsack problems." *Proceedings of the 18th Annual Symposium on Foundations of Computer Science.* IEEE Computer Society, Long Beach, CA (1977). pp. 206-213.

[5] Papadimitriou, C. H. "The NP-Completeness of the bandwidth minimization problem." *Computing* 16 (1976). pp. 263-270.

[6] Shamos, M. I. Personal communication (1978).

[7] Yemini, Y. "The positioning problem--A draft of an intermediate summary." In *Distributed Sensor Nets* [2].

FAST RECOGNITION OF BAXTER PERMUTATIONS USING SYNTACTICAL AND COMPLETE BIPARTITE COMPOSITE DAGS

Preliminary Version

JOHNSON M. HART
University of Kentucky, Lexington, Kentucky 40506

ABSTRACT

A correspondence is developed between permutations and two-dimensional digraphs. When the graphs are also required to be complete bipartite composite (CBC), their permutations then correspond to a class called the Baxter permutations. A permutation can be tested for the Baxter condition by a linear time (in the number of vertices) processing of its digraph. This processing involves a pair of traversals of the digraph and is made possible by exploiting a relationship with the syntactical graphs used in language theory.

INTRODUCTION

Two dimensional digraphs, such as the syntactical graphs [1,6,9] used in language theory and the Complete Bipartite Graphs of Valdes et al. [11,12] can be represented by permutations. When the graphs are also Complete Bipartite Composite (CBC), the corresponding permutations turn out to be the so-called Baxter permutations which occur in functional analysis, especially in the study of commuting functions [2-4]. In these studies, the generation and recognition of Baxter permutations is important.

In this paper, we improve the obvious $O(n^2)$ recognition algorithm [3] to obtain an $O(n)$ technique. The new algorithm requires the indirect construction of the two-dimensional graph and testing for the CBC property in a manner similar to that used in [11,12] to recognize General Series Parallel graphs in order to obtain efficient algorithms for scheduling under constraints. In the process, the CBC graph is transformed into a graph very much like the syntactical graphs used to represent phrase structure derivations [1,6] and as the domain of graph automata [6,9]. The resulting graph is processed as in [6] to yield the final recognition of the Baxter permutations.

DAG'S, DOG'S, AND TWO-DIMENSIONAL GRAPHS

As is standard, a directed graph (digraph) $G = \langle V,E \rangle$ is a finite set of vertices, V, and a finite set of edges, E, where each edge is an ordered pair of vertices.

The usage of the terms path, cycle, source, and sink in a digraph are standard, as is dag (directed acyclic graph).

It is convenient to regard E as a relation on V with E^+ (E^*) the transitive (and reflexive) closure of E. The transitive reduction of

$G = \langle V,E\rangle$ is $G_0 = \langle V,E_0\rangle$ where E_0 is the smallest relation such that $E^+ = E_0^+$. G is *minimal* if $G = G_0$.

A dag $G = \langle V,E\rangle$ is two-dimensional [11,12] (more strictly, the partial order E^* is 2D) if and only if there are two total orders $<_1$ and $<_2$ on V such that $E^* = <_1 \cap <_2$. That is, there is a path from u to v just in case $v <_1 u$ and $v <_2 u$. A 2D partial order is similarly defined.

The above definitions are standard from graph theory. The next definition (of a DOG) was developed [1] to simplify the description of derivation structures.

Let B and L be two transitive relations on V (finite) with $B^r = \{(x,y) \mid (y,x) \varepsilon B\}$ (the reversal of B; similarly for L^r). Then $D = \langle V,B,L\rangle$ is a *Doubly Ordered Graph* (DOG) just in case

$$\{B,L,B^r,L^r,=\}$$

is a partition of V^2 (i.e., every pair of distinct vertices is related by just one of B or L or their reversals).

It follows immediately that any DOG forms two distinct dags over V; namely, one with B and one with L. $B \cup \{=\}$ and $L \cup \{=\}$ are both partial orders on V.

The term *ordered dag* is sometimes used for constructs similar to DOG's. Usually, however, equality is included in the B relation.

Interpreting B as "below" and L as "left of", a tree (where left-to-right sibling order is important) is a special case of a DOG. Also, the graph shown in Fig. 1 where the transitive reduction of B is shown and L is inferred by position, is a typical DOG. The reason for the sequential labelling of the vertices will become apparent.

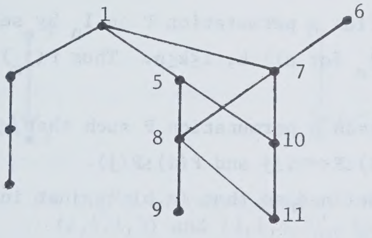


Fig. 1. A Typical Doubly Ordered Graph

Lemma 2.1. A partial order B on V is two-dimensional if and only if there is a relation L such that $\langle V,B',L\rangle$ is a DOG, where $B' = B-\{=\} = (B_0)^+$.
Proof: a) *Only if*. Let B be 2D with total orders $<_1$ and $<_2$ such that $B = <_1 \cap <_2$. Set

$$L = \{(x,y) \mid x <_1 y \text{ and } y <_2 x\} - \{(x,x) \mid x \varepsilon V\}.$$

Then clearly $\{B', B'^r, L, L^r, =\}$ is a partition of V^2 and L is transitive. Here, $B' = (B_0)^+$.

b) *If*. Suppose that $D = \langle V, B', L \rangle$ is a DOG for some relation L. Define the relations $<_1$ and $<_2$ as

$$<_1 \; = B \cup L, \text{ and}$$
$$<_2 \; = B \cup L^r.$$

Certainly, each of the < is total and $B = \, <_1 \cap <_2$. Each order is also easily shown to be transitive. □

Considering the example in Figure 1, the vertices are numbered in the reverse order of $<_2 = B \cup L^r$. Hence $11 <_2 10 <_2 9$, etc. The $<_1$ order $(B \cup L)$ is:

4,3,2,9,11,8,10,5,7,6,1.

Let $I_n = \{1,2,\ldots,n\}$ for all $n \geq 1$. A partial order E on I_n is said to be *consistent* if $(i,j) \varepsilon E$ implies that $i \leq j$ so that the partial order does not conflict with numerical order. E is *strictly 2D* if $E = \, \leq \cap <_1$ for some total order $<_1$, where $\leq$ is numerical order. There is no loss of generality in requiring all partial orders on I_n to be consistent and 2D p.o.'s to be strict, for the elements can be relabelled to conform to one of the two total orders. The correspondence with permutations can now be established.

Lemma 2.2. A consistent partial order E on I_n is strictly 2D iff there is a permutation P on I_n such that

$$(i,j) \varepsilon E \Longleftrightarrow i \leq j \text{ and } P(i) \leq P(j).$$

Proof: a) *Only if*. Suppose that E is strictly 2D with $E = \, \leq \cap <_1$. Write the elements of I_n in $<_1$ order as

$$j_1, j_2, \ldots, j_n$$

so that $(j_k, j_{k+1}) \varepsilon <_1$. Define a permutation P on I_n by setting $P(k) = m$ where m is such that $k = j_m$ for all k, $1 \leq k \leq n$. Thus $P(j_m) = m$ suffices for the lemma.

b) *If*. There exists a permutation P such that

$$(i,j) \varepsilon E \Longleftrightarrow i \leq j \text{ and } P(i) \leq P(j).$$

Let the total order $<_1$ be defined so that $(a,b) \varepsilon <_1$ just in case $P(a) \leq P(b)$. $E = \, \leq \cap <_1$, and E is strictly 2D. □

In keeping with these results, for any permutation, P, define the DOG, D_p, by

$$D_p = \langle I_n, B, L \rangle$$

where

1) $(i,j) \varepsilon B \Longleftrightarrow i > j$ and $P(i) > P(j)$, and
2) $(i,j) \varepsilon L \Longleftrightarrow i < j$ and $P(i) > P(j)$.

As a result of this, the DOG's and the permutations correspond exactly, and they both correspond with the 2D partial orders (up to a relabelling to give strictness).

Example 2.1. Fig. 1 represents a DOG. The corresponding permutation is

$$P_1 = (2,9,10,11,4,1,3,6,8,5,7).$$

The reason for the labelling of the vertices in Fig. 1 should now be apparent.

BAXTER PERMUTATIONS AND THEIR FORBIDDEN SUBGRAPH CHARACTERIZATION

Strictly speaking, Baxter permutations (BP) are defined for all odd values of $n \geq 7$, but a Baxter Permutation is determined entirely by its values on the odd numbers [2,3]. Hence we will examine permutations on all values of n and call a permutation a BP (or a reduced BP [5]) just in case the permutation induced on $\{1,2,\ldots,2n-1\}$ is a BP in the strict sense [2,3]. Let $I_n = \{1,\ldots,n\}$ and we then have

Definition 3.1. A permutation $P: I_n \rightarrow I_n$ is a Baxter permutation of order n if and only if there are no integers i, j, and k such that

$$i<j-1<j<k \text{ and either}$$

$$P(j)<P(i) = P(k)-1<P(j-1), \text{ or}$$

$$P(j-1)<P(k) = P(i)-1<P(j).$$

This definition is different from those in [5,8] but is easily shown to be equivalent.

There are only two non-BP's of order four, namely (2,4,1,3) and (3,1,4,2). As DOG's, these two non-BP's are shown in Fig. 2 where the transitive reduction of B is shown and the L relation is inferred by position.

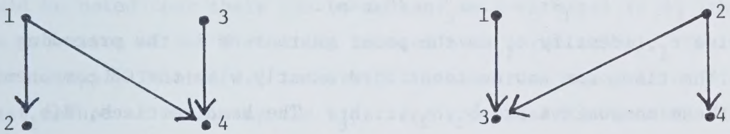


Fig. 2. The DOG's corresponding to the non-BP's (2,4,1,3) and (3,1,4,2), resp.

More generally, a permutation is a BP if and only if its DOG (in the transitive reduction) does not contain one of these two forbidden subgraphs (called N's) [8]. This forbidden N subgraph also appears in [11,12] as a forbidden subgraph in a minimal series parallel graph. The existence of an N clearly causes the permutation to be non-Baxter, and the violation of the Baxter condition shows up as an N when the DOG is drawn.

A digraph $G = \langle V,E \rangle$ is complete bipartite if V can be partitioned into H and T so that $E = H \times T$. H is the head and T is the tail of G.

A dag G is Complete Bipartite Composite (CBC) [11,12] if there exists a set of complete bipartite subgraphs of $G: b_1, b_2, \ldots, b_k$ (the bipartite components of G) such that:

a) each edge of G belongs to exactly one bipartite component,

b) every vertex v of G, except the sinks, belongs to the head of exactly one bipartite component denoted h(v), and

c) every vertex v of G, except the sources, belongs to the tail of exactly one bipartite component, denoted t(v).

From this, a characterization of BP's can be inferred (and proved formally) using the forbidden N subgraph characterization.

Theorem 3.1. A permutation P on I_N is a BP if and only if its DOG $D_p = \langle I_N, B, L \rangle$ is CBC w.r.t. the transitive reduction of B.

Proof: a) If. If P is not a BP, then some N must exist. The vertices in this N violate the conditions of the CBC definition.

b) Only if. Assume that P is a BP on I_n with $D_p = \langle I_n, B, L \rangle$ the corresponding DOG. Using the statements of the Baxter condition (Def. 3.1) (which states which configurations cannot exist), whenever

$$P(j-1)<P(j)$$

there exists a unique t, $P(j-1) \le t < P(j)$ such that

a) $P(j-1) \le s \le t \Rightarrow P^{-1}(s) \le j-1$, and

b) $t < s \le P(j) \Rightarrow P^{-1}(s) \ge j$.

An index j satisfying the condition that $P(j-1)<P(j)$ is called a rise of P. Write all of the rises in increasing order as

$$r_1 < r_2 < \ldots < r_c \quad (0 \le c < n).$$

With each rise r_i, identify t_i as the point guaranteed in the preceding paragraph. The rises can now be identified exactly with the CB components of D_p, and these components are $b_1, b_2, \ldots, b_c$. The head vertices, $H(b_i)$ or simply H_i, and the tail vertices, T_i, are defined inductively. Use the notation

$$X_i = \bigcup_{j=1}^{i-1} H_i \quad \text{and} \quad Y_i = \bigcup_{j=i+1}^{c} T_j$$

Then, for $1 \le i \le c$,

$$H_i = P^{-1}(\{P(r_i-1), \ldots, t_i\}) - X_i, \text{ and}$$

$$T_i = P^{-1}(\{t_i+1, \ldots, P(r_i)\}) - Y_i.$$

Since r_1 is the first rise, $(r_1, x) \in B_0$ if and only if $x \in H_1$, and, in addition, if $x<y$ with $x, y \in H_1$, then $(x,y) \in L$. The elements of H_1 are

consecutive in I_n and each element is immediately to the left of its successor.

Restricting P to the domain $I_n - X_2$, r_2 is the first rise of the resulting function, and we likewise have $(r_2, x) \varepsilon B_0$ iff $x \varepsilon H_2$. Continuing this process, restricting P to $I_n - X_i$,

$$(r_i, x) \varepsilon B_0 \text{ iff } x \varepsilon H_i \quad (1 \leq i \leq c),$$

and the elements of H_i are consecutive in $I_n - X_i$ with $(x,y) \varepsilon L_0$ if x and y are consecutive in $I_n - X_i$.

By an entirely symmetrical argument

$$(x, r_{i-1}) \varepsilon B_0 \text{ if } x \varepsilon T_i \quad (1 \leq i \leq c),$$

and the elements of T_i are numerically in L_0 order. In addition, from the construction and the role played by the t_i, it is clear that

$$T_i \times H_i \subseteq B \quad (1 \leq i \leq c),$$

and it can be shown that $T_i \times H_i \subseteq B_0$.

Finally, it is clear that every B_0 relation (edge of D_p) must be in some component, so

$$B_0 = \bigcup_{i=1}^{c} T_i \times H_i,$$

which completes the theorem. □

This theorem and its proof are the basis for the recognition algorithm of the next section.

RECOGNITION OF BAXTER PERMUTATIONS

Boyce [3] tests for Baxter permutations by the obvious brute force technique using an $O(N^2)$ algorithm. Valdes et al. [11,12] give a linear (in # of edges + # of vertices) algorithm for the CBC property, but it should be noted that there can be an many as $N^2/4$ edges in B_0 (the transitive reduction of the B relation induced by a permutation). Therefore, in order to apply Theorem 3.1 to obtain a linear time recognition of BP's, a faster method of obtaining the bipartite components is needed. The basic idea is suggested by the syntactical graphs and derivation words [6] which are also known to be DOG's [1], as should be expected by the fact that syntactical graphs generalize context-free derivation trees and their prefix representation.

Given a bipartite component b of a CBC graph with edges $H \times T$ (i.e., mn edges where $m = \# H$ and $n = \# T$) add a new vertex b and replace the edges of b with $H \times \{b\} \cup \{b\} \times T$, for a total of m+n edges and one new vertex per component. The transformation is shown in Figure 3.

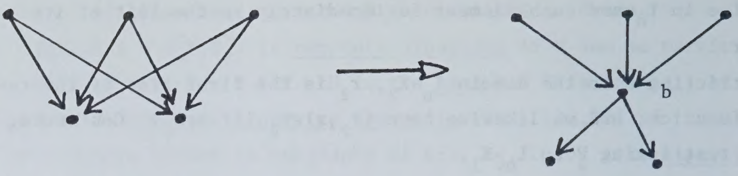


Fig. 3. The Transformation of a Bipartite Component

The proof of Theorem 3.1 implies an algorithm to compute the head and tail vertices of each component, where a component corresponds to a rise of the permutation. Suppose that this algorithm has been carried out so that for each component b_i, the sets H_i and T_i are represented by doubly linked circular lists (the technique for achieving this will be demonstrated later in this section). The lists are also ordered according to L and vertices can be removed from the lists in either L order or L^r order.

In what follows, it is convenient to talk of $B \cup L$, $B \cup L^r$, etc. as total orders (See the proof of Lemma 2.1). These orders are not quite total since the equality relation is omitted, but no confusion can result from this.

Now, the rises are ordered by $B^r \cup L$; that is, if $i<j$, then $(r_i, r_j) \varepsilon B^r \cup L$. In fact, the same can be said of the head and tail elements so that if $i<j$ and $x \varepsilon H_i \cup T_i$ and $y \varepsilon H_j \cup T_j$, we have $(x,y) \varepsilon B^r \cup L \cup =$.

The sources of the DOG are available in $B^r \cup L$ or $B^r \cup L^r$ order as $I_n - Y_0 = I_n - \bigcup_{j=1}^{c} T_j$. From all this, it is clear that we can use the adjacency lists of the components and the list of sources to perform a standard depth first search (in B^r order) of the DOG [10, pp 292-293 or any data structures text]. If the vertices are removed from the adjacency lists in L^r order, then the traversal will be in $B^r \cup L^r$ order.

By definition $(i,j) \varepsilon B^r \cup L^r$ if and only if $P(i)<P(j)$. Let the order of traversal obtained in this way be

$$i_1, i_2, \ldots, i_n.$$

Perform a second traversal in $B^r \cup L$ order so that $(i,j) \varepsilon B^r \cup L$ iff $i<j$. Let the sequence obtained be

$$j_1, j_2, \ldots, j_n.$$

This should be the numerical sequence but can fail to be if the permutation is not Baxter. If it is not numerical, halt. Otherwise create a permutation Q such that $Q(i_k)=k$. If the traversals were performed correctly, then P=Q.

Depth first search is fast (linear time in n). If the algorithm for constructing the H_i and T_i can be shown to be fast and if $P \neq Q$ when P is not Baxter, then the fast algorithm for recognizing Baxter permutations will be complete.

The fast computation of the H_i and T_i is based upon the use of a push-down stack. Regard the permutation as the string of numbers, 1,2,...,n in order with the values of P available as a function. Scan the string from 1 to r_1-1, pushing the numbers onto a stack (initially empty). Then pop from the stack until the top of stack element (called s) is such that $P(s)>P(r)$ or the stack is empty, leaving s on the stack (a nondestructive read is assumed). As elements are popped from the stack, place them on the H_1 adjacency list, noting that they are obtained in L^r order. Since r_1 is the first rise, the H_1 adjacency list will be complete.

Resume the scan until r_2-1, which represents the first rise of the restricted function since the elements of H_1 have been removed. Repeat this process until n is scanned. The sink vertices will be the only ones left on the stack, where they will appear, from top-to-bottom, in L^r order.

A symmetrical process, scanning from n to r_c and on to 1, obtains the T_i and the source vertices. The stack is popped until s (the TOS) is such that $P(s)<P(r-1)$ or the stack is empty.

This algorithm works correctly if P is a BP. If P is not a BP, however, the situation is somewhat different. The operation of the algorithm is such that

$$H_i \cap H_j = T_i \cap T_j = \emptyset \quad \text{if } i \neq j.$$

That is, every vertex belongs to one head (or tail) at most. The resulting structure represents a CBC 2D graph, and when the two traversals are performed as indicated above, the resulting permutation, Q, must be a BP and cannot be equal to P.

In summary, the recognition algorithm for permutation P works as follows:

1) For each rise, r_i, compute H_i and T_i.
2) Perform the $B^{-r} \cup L^r$ traversal to obtain the order

$$i_1, i_2, \ldots, i_n.$$

3) Perform the $B^r \cup L$ traversal to obtain the order

$$j_1, j_2, \ldots, j_n.$$

If this sequence is not numerical (i.e., $i_k \neq k$ for some k), then halt since P is not a BP.

4) Otherwise, compute the permutation Q such that

$$Q(i_k)=k \quad \text{for } 1\le k\le n.$$

5) If $P\neq Q$, then P is not Baxter, otherwise it is.

REFERENCES

[1] R. Alter and J. M. Hart, Enumerating syntactical graphs and lattices of derivations, International Journal of Computer and Information Sciences 8, 4 (1979), 261-277.

[2] G. Baxter, On fixed points of the composite of computing functions, Proc. Amer. Math. Soc. 15 (1964), 851-855.

[3] W. M. Boyce, Generation of a class of permutation associated with commuting functions, Math. Algorithms 2 (1967), 19-26.

[4] W. M. Boyce, Baxter permutations and functional composition, to appear.

[5] F. R. K. Chung, R. L. Graham, V. E. Hoggatt, and M. Kleiman, The number of Baxter permutations, J. of Combinatorial Th. (Series A) 24 (1978), 382-394.

[6] J. M. Hart, Derivation languages and syntactical categories, Inform. Contr. 28 (1975), 204-220.

[7] J. M. Hart, Doubly keyed lists for binary relations, 1979 CISS, The Johns Hopkins University, Baltimore, MD.

[8] J. M. Hart and R. Alter, Counting Baxter Permutations with Doubly Ordered Graphs (1978), TR#46-78, University of Kentucky, Lexington, KY (Dept. of Computer Science).

[9] T. Kamimura and G. Slutzki, Dags and Chomsky hierarchy, 1979 ICALP, Graz, Austria, to appear.

[10] E. Horowitz and S. Sahni, Fundamentals of Data Structures (1976), Computer Science Press, Woodland Hills, CA.

[11] J. Valdes, Parsing Flowcharts and Series - Parallel Graphs (Aug. 1978), Ph.D. dissertation, Department of Computer Science, Stanford University.

[12] J. Valdes, R. E. Tarjan, and E. L. Lawler, The recognition of series parallel digraphs, 1979 Eleventh Annual Symposium on Theory of Computing (Proceedings) 1-12.

Dynamic-Programming Algorithms for Recognizing Small-Bandwidth Graphs in Polynomial Time[1]

JAMES B. SAXE
Computer Science Department
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

Abstract--In this paper we investigate the problem of testing the bandwidth of a graph: Given a graph, G, can the vertices of V be mapped to distinct positive integers so that no edge of G has its endpoints mapped to integers which differ by more than some fixed constant, k? We exhibit an algorithm to solve this problem in $O(f(k)N^{k+1})$ time, where N is the number of vertices of G and f(k) depends only on k. This result shows that the "Bandwidth(G) $\overset{?}{\leq}$ k" problem is not NP-Complete (unless P = NP) for any fixed k, answering an open question of Garey, Graham, Johnson, and Knuth.

1. Introduction

The subject of this paper is the computational complexity of a problem on graphs. To speak precisely of the problem, we will need the following notation and definitions.

Notation:

Let u and v be vertices of a graph G. We will say "u—v in G" to denote that {u,v} is an edge of G. Where G is clear from context, we will write simply "u—v".

Definitions:

Let G be a graph with vertex set V, and let N = |V|. A layout of G is a one-to-one mapping, f, from V onto {1,...,N}. The bandwidth of f is defined as the maximum distance between the images under f of any two vertices that are connected by an edge of G. That is,

$$\text{bandwidth}(f) = \max\{f(u)-f(v) \mid u—v\}.$$

The bandwidth of G is defined as the least possible bandwidth for any layout of G. Thus

$$\text{Bandwidth}(G) = \min\{\text{bandwidth}(f) \mid f \text{ is a layout of } G\}.$$

Problem (Bandwidth Minimization):

Given an arbitrary graph, G, and a positive integer, k, determine whether Bandwidth(G) ≤ k.

---

[1] This research was supported in part by the Office of Naval Research under Contract N00014-76-C-0370.

Note that the notion of graph bandwidth is equivalent to the more familiar notion of matrix bandwidth in that Bandwidth(G) ≤ k iff there exists a permutation matrix P such that $(PCP^{-1})_{i,j} = 0$ whenever $|i-j| > k$, where C is G's connection matrix. For any particular positive integer k, we can define a restricted version of the bandwidth minimization problem as follows:

Problem (Bandwidth $\overset{?}{\leq}$ k):
Given a graph, G, determine whether Bandwidth(G) ≤ k.

Papadimitriou [5] has shown that the general bandwidth minimization problem, in which k is specified in the input, is NP-Complete. The problem was later studied by Garey, Graham, Johnson, and Knuth [2], who found a linear-time algorithm for the the problem "Bandwidth $\overset{?}{\leq}$ 2", and also improved on Papidimitriou's result by showing the problem for general k to be NP-Complete even when G is restricted to be a tree with no vertex of degree greater than three. A number of questions are left open by their work, however. One such question is whether there exists a polynomial-time algorithm for the problem "Bandwidth $\overset{?}{\leq}$ 3". In this paper, we will answer this question affirmatively by exhibiting an algorithm[2] which solves the problem "Bandwidth $\overset{?}{\leq}$ k" in polynomial time for any fixed k. Section 2 of this paper introduces the fundamental concepts and assumptions we will use in describing our algorithm. In Section 3 the algorithm is described and its performance is analyzed. In Section 4 we discuss some modifications of the algorithm to solve related problems. Finally, in Section 5, we discuss some remaining open problems.

2. Fundamental Concepts and Assumptions

Throughout the following we will assume that G denotes a graph with vertex set V and edge set E, that k denotes a particular positive integer,[3] and that we wish to determine whether G has any layout of bandwidth ≤ k. We let N denote the cardinality of V. Note that if G is not connected then G has a layout of bandwidth ≤ k iff each of its components has such a layout. Also, it is clearly impossible for G to have such a layout if G has any vertex of degree 2k or greater. We therefore assume, without loss of generality, that G is a connected graph having no vertex of degree greater than or equal to 2k. Note that an arbitrary graph can be partitioned into its connected components by depth-first search in O(max(n,e)) time, where n is the number of vertices and e is the number of edges,[4] and that this is O(n) if a fixed bound is given on the degree of any vertex. Moreover, an obvious modification to the depth-first search algorithm allows it to detect the presence of a vertex with degree greater than a fixed bound in time which is proportional only to the number of vertices and not to the number of edges.

---

[2] More correctly, a class of algorithms, one for each value of k.

[3] When using the "big-oh" notation, we will regard k as fixed and therefore omit factors that depend only on k.

We now introduce the key notion of a partial layout.

Definitions:

A partial layout of G is a one-to-one function, f, from some subset of V onto $\{1,...,M\}$, for some M such that $0 \le M \le N$.. We say that f is feasible if it can be extended to a (total) layout, g, such that bandwidth(g) $\le k$. The bandwidth of f is the maximum distance between the images of any two edge-connected vertices of G which are in the domain of f. If u—v and u is in the domain of f and g is not, then the edge $\{u,v\}$ is said to be dangling from f.

Consider a partial layout, f, of size M. Clearly, f cannot possibly be feasible unless

1. bandwidth(f) $\le k$, and
2. whenever u and v are vertices of G such that $f(u) < M-k$ and u—v, v is also in the domain of f.

If f satisfies both these conditions, then f is said to be a plausible partial layout. The sequence $(f^{-1}(\max(M-k+1,1)),...,f^{-1}(M))$, taken together with the set of dangling edges of f, is called the active region of f. We now come to the theorem on which our principal algorithm depends.

Theorem 2.1:

Let f and g be two plausible partial layouts of G having identical active regions. Then,

1. f and g have identical domains, and
2. f is feasible iff g is feasible.

Proof:

Since G is connected, the domains of f and g must each consist precisely of those vertices which are path-connected to vertices in the active region by paths not including any dangling edges. Thus, (1) holds. To see that (2) holds, we need only note that any assignment of the remaining vertices which extends either f or g to a total layout of bandwidth $\le k$ must also extend the other to such a layout. □

Finally, we define the notion of a successor of a plausible partial layout (or active region), which will be necessary to explain our algorithms.

---

[4] See, for example, Aho, Hopcroft, and Ullman [1, Chapter 5].

Definition:

Let f be a plausible partial layout of G. Then a successor of f is a plausible partial layout, g, which extends f by precisely one element. In this case, the active region of g is also said to be a successor of the active region of f. We also say that (the active region of) f is a predecessor of (the active region of) g.

3. The Algorithm

Theorem 2.1 allows us to say that two plausible partial layouts are equivalent if they have identical active regions. The algorithm we present is essentially a breadth-first search over the space of all the induced equivalence classes of plausible partial layouts, where each such equivalence class is uniquely characterized by active region of its representatives. Alternatively, we may think of the algorithm as a dynamic-programming search over the plausible partial layouts. Each active region consists of at most k vertices and each vertex has no more than 2k edges, each of which may or may not be dangling. Thus the number of equivalence classes is bounded above by[5]

$$\sum_{0\leq i\leq k} \binom{N}{i}(2^{2k})^i = O(N^k).$$

Our algorithm uses the following two data structures:

1. A (fifo) queue, Q, whose elements are active regions.
2. An array, A, which contains one element for each possible active region. Each element, A[r], of A consists of a Boolean flag, A[r].examined, telling whether the active region r has already been considered in the search and a list, A[r].unplaced, of vertices which is intended to list all vertices NOT in the domain of each plausible partial layout with active region r.

At the start of our algorithm, Q is initialized to contain the single element representing the active region (henceforward denoted Φ) of the empty partial layout, ϕ. The flag A[Φ].examined is set to TRUE and A[Φ].unplaced is initialized to list all the elements of V. The remaining A[r].examined are initially FALSE, and the remaining A[r].unplaced are uninitialized. The algorithm now proceeds as follows:

---

[5] As we will mention in Section 5, the coefficient on this bound is quite loose.

Algorithm B (Bandwidth testing):

```
1. Extract an active region, r, from the head of Q.
2. From A[r].unplaced, determine the successors of r.
3. For each successor, s, of r such that A[s].examined is FALSE, perform the
   following steps:
      a. Set A[s].examined to TRUE.
      b. Compute A[s].unplaced by deleting the last vertex of s from
         A[r].unplaced.
      c. If A[r].unplaced is the empty set, then halt asserting that
         Bandwidth(G) $\le$ k.
      d. Insert s at the end of Q.
4. If Q is empty, then halt asserting that Bandwidth(G) $>$ k. Otherwise, go to
   Step 1.
```

The space required by this algorithm is clearly $O(N^{k+1})$. To determine the running time, we note first that since there are $O(N^k)$ possible active regions, each of Steps 1 through 4 will be executed $O(N^k)$ times. The individual executions of Steps 1 and 4 each take only constant time, so the contribution of these steps to the total running time of the algorithm is $O(N^k)$. Since any active region, r, has at most N successors (zero or one for each element of A[r].unplaced), each execution of Step 2 takes O(N) time. The contribution of Step 2 to the total execution time is therefore $O(N^{k+1})$. Determining the contribution of Step 3 is (a little) trickier. During a single execution of Step 3, Steps 3.a through 3.d may be executed as many as N times, and the amount of computation in Step 3.b may be $\theta(N)$. Thus it appears possible that Step 3 may contribute $\theta(N^{k+2})$ to the total execution time. If we look more carefully, however, we see that 3.a through 3.d are executed at most once for each active region. Thus the total contribution of Step 3 is $O(N^{k+1})$. Adding the contributions of all the steps gives us the following result.

Theorem 3.1:

Let k be any positive integer. Then there is an algorithm which solves the problem "Bandwidth $\overset{?}{\le}$ k" using $O(N^{k+1})$ time and $O(N^{k+1})$ space.

Proof:

To test the bandwidth of G, we first perform an O(N)-time depth first search which either

(1) determines that G has some vertex of degree greater than 2k, or

(2) partitions G into connected components none of which have any vertex of degree greater than 2k.

In case (1), we know immediately tnat Bandwidth(G) ≥ k. In case (2), we apply Algorithm B to the connected components of G. ☐

While Algorithm B will tell us whether G has a layout of bandwidth ≤ k, it does not actually produce such a layout. In order to allow such a layout to be recovered, we may associate with each active region, s, an additional field, A[s].predecessor. When s is appended to Q in Step 3.f., we make A[s].predecessor point to a predecessor of s (namely the r we chose in Step 1).[6] If the algorithm finds an active region, t, such that A[t].unplaced is empty, it is a simple matter to recover a layout by tracing back through the predecessor fields.

4. Modifications for Related Problems

Another question left open by Garey, Graham, Johnson, and Knuth [2] is whether there exists a polynomial-time algorithm to *count* the layouts of a graph having bandwidth ≤ k, even for k = 2. We now give an affirmative answer to a closely related question by exhibiting a class of polynomial-time algorithms (one for each positive integer k) for determining the number of bandwidth ≤ k layouts of any *connected* graph.[7]

Our algorithm for enumerating layouts of bandwidth ≤ k is a slightly modified form of Algorithm B. The data structures are the same as those for Algorithm B, with the following additions:

1. Each entry, A[r], of A has a third field, A[r].count, which will hold the number of (so far discovered) plausible partial layouts whose active region is r.
2. There is a variable, Total, which will hold the number of (so far discovered) layouts of bandwidth ≤ k.

At the start of the algorithm, Total and all the A[r].count are initialized to zero, except for A[Φ], which is initialized to 1. The remaining variables are initialized as for Algorithm B. We then proceed as follows:

Algorithm E (Enumerate layouts):

```
1. Extract an active region, r, from the head of Q.
```

---

[6] Note that this pointer need only name the single vertex (if any) which is contained in r but not in s.

[7] Note that the number of bandwidth ≤ k layouts of an arbitrary graph is not uniquely determined by the numbers of bandwidth ≤ k layouts of its connected components because the topologies of the components impose constraints on how the various layouts may overlap. The algorithms cannot be applied directly to non-connected graphs because it depends on Theorem 2.1.

2. From A[r].unplaced, determine the successors of r.
3. For each successor, s, of r, perform the following steps:
    a. If A[s].examined is TRUE, go to f.
    b. Set A[s].examined to TRUE.
    c. Compute A[s].unplaced by deleting the last vertex of s from A[r].unplaced.
    d. If A[r].unplaced is the empty set, then increase Total by A[r].count.
    e. Insert s at the end of Q.
    f. Increase A[s].count by A[r].count.
4. If Q is empty, then halt. Otherwise, go to Step 1.

Study of this algorithm gives us the following result:

Theorem 4.1:

Let k be any positive integer. Then there exists an $O(N^{k+1})$-time, $O(N^{k+1})$-space algorithm which, given any *connected* graph, G, computes the number of layouts of G having bandwidth $\le k$.

Proof:

We claim that Algorithm E (preceded by a depth-first search to ensure that no vertex of G has degree greater than 2k) has the desired properties. By an analysis similar to that for Algorithm B, Algorithm E will run in $O(N^{k+1})$ time. We must now show that it correctly counts the layouts of bandwidth $\le k$. To do this, it suffices to show that by the time that any plausible partial layout, r, is selected in Step 1, A[r].count contains the total number of plausible partial layouts whose active region is r. This in turn may be shown inductively if we can only show that no active region, r, is chosen in Step 1 until every predecessor of r has been chosen. This last follows at once from the fact (which may be established by induction) that the active regions proceed through the queue in non-decreasing order of their lengths, where the length of an active region, r, is defined to be the number of vertices in the domain of any plausible partial layout whose active region is r. ☐

We may view Bandwidth Minimization as the problem of finding a layout with minimax edge length. We will now look at the corresponding minisum problem.

Definition:

Let G be a graph with edge set E, and let f be a layout of G. Then the total edge length of f is given by the sum

$$\sum_{\{u,v\}\in E} |f(u)-f(v)|$$

where each edge, {u,v}, contributes precisely once to the sum (rather than once as u—v and once as v—u).

Problem: (Optimal Linear Arrangement)
Given a graph, G, and an integer, t, determine whether there is a layout of G having total edge less than or equal to t.

The Optimal Linear Arrangement (O.L.A.) problem was found to be NP-Complete by Garey, Johnson, and Stockmeyer [4]. However, Shamos [6] has pointed out that the methods of the present work can be used to provide polynomial-time algorithms for a class of restricted versions of O.L.A. For every positive integer, k, we define a restriction of O.L.A. as follows:

Problem: (O.L.A. for bandwidth $\leq$ k)
Given a graph, G, determine the minimal total edge length of any layout of G having bandwidth $\leq$ k or determine that no such layout exists.

Applying the methods used above, we obtain the following result.

Theorem 4.2:
Let k be any positive integer. Then there exists an algorithm which solves O.L.A. for bandwidth $\leq$ k in $O(N^{k+1})$ time and $O(N^{k+1})$ space.

Proof:
An algorithm having the desired properties when applied to connected graphs with no vertex having degree greater than 2k may be constructed by a slight modification of Algorithm E: instead of maintaining with each active region a count of the partial layouts having that active region, we maintain an indication of the minimum sum of the lengths of all edges whose endpoints have are in the domains of all plausible partial layouts having that active region. The details are left to the reader. For arbitrary graphs we first perform a depth-first search which either detects the presence of a vertex with degree greater than 2k (implying that Bandwidth(G) $\geq$ k) or partitions G into its connected components, taking linear time in either case. We then compute the minimal total edge length for G by finding and summing the minimal total edge lengths for the connected components. ☐

We note that the previous result remains valid if we consider edge weighted graphs and the "total edge length" is taken as a weighted sum. For connected graphs, we can also use the method of Algorithm E to obtain a count of the layouts with minimal total edge length for bandwidth $\leq$ k.

Finally, all the previous results extend to "directed" versions of Bandwidth

Minimization and O.L.A., in which G is a directed graph and a layout, f, is acceptable only if $f(u) < f(v)$ whenever (u,v) is an edge of G.[8]

5. Open Problems

The most obvious problem left open by this work is that of improving the performance of Algorithm B. Although the expense of this algorithm is "only polynomial" in the size of the examined graph it is still sufficiently expensive (particularly in terms of space) to render it impractical for all but the smallest cases (consider, for example, determining whether Bandwidth(G) ≤ 5, where G is a graph of forty vertices). The fact that Garey, Graham, Johnson, and Knuth [2] have a linear-time algorithm for "Bandwidth(G) $\overset{?}{\leq}$ 2", while Algorithm B takes cubic time for the same problem offers some hope that the degree of the polynomial can be reduced for higher values of k as well. Indeed, it is conceivable (even if P ≠ NP) that there are linear algorithms for all values of k, with coefficients growing exponentially in k.

One approach to improving the performance is to attempt to reduce the number of active regions examined, and this can indeed be done to some extent. For example, we may prune the search by noting that, while a plausible partial layout may have $\theta(k^2)$ dangling edges, such a partial layout cannot actually be feasible if those edges lead to more than k distinct vertices. Unfortunately, graphs of the form

$$v_1 — v_2 — \dots — v_{N-1} — v_N$$

supply an existence proof that the number of equivalence classes of plausible partial layouts of bandwidth ≤ k can in fact be $\theta(N^k)$.

In Algorithm B, we reduce the search space from the set of all plausible partial layouts to the much smaller set of equivalence classes of partial layouts. To look at it another way, given two partial layouts, f and g, if we recognize (by equality of active regions) that f is feasible iff g is feasible, then we feel free to search for completions of only one of the partial layouts. The algorithm of Garey, Graham, Johnson, and Knuth cuts down the search space by methods which are similar but more sophisticated. In particular, they can avoid searching for completions of a partial layout,[9] f, by choosing to search for completions of a layout, g, such that g is feasible whenever f is feasible, but not necessarily only when f is feasible.

---

[8] A good starting point for the reader who is interested in learning more about Bandwidth, O.L.A., and their variations is Appendix A1 of Garey and Johnson [3]

[9] In their terminology, a partial layout of G is a map from a subset of the vertices of G to an *arbitrary* set of integers.

It is interesting to note that "worst-case" numbers of feasible active regions seem to arise precisely in circumstances where large pieces of the graph can be laid out in bandwidth much less than k. We define a maximal graph of size N and bandwidth k as a graph whose edge set is $\{\{v_i,v_j\} \mid |i-j| \leq k\}$, where $\{v_i \mid 1 \leq i \leq N\}$ is the edge set.[10] The algorithm of Garey, Graham, Johnson, and Knuth relies heavily on the fact that if all the even numbered vertices or all the odd numbered vertices are deleted from a maximal graph of bandwidth 2, the induced graph on the remaining vertices is a maximal graph of bandwidth 1. For testing higher bandwidths it is possible that similar use may be made of the fact that deleting every k-th vertex from a maximal graph of bandwidth k leaves a maximal graph of bandwidth k-1.

Another potentially fruitful course of investigation would be to look for efficient algorithms for approximate bandwidth minimization. For example, given a graph, G, we may wish to produce a layout for G whose bandwidth is no more than, say, twice the minimum possible. To the author's knowldge it has not yet been determined whether this problem (when phrased as a language recognition problem) is NP-Complete.

Acknowledgements

The author gratefully acknowledges the helpful comments of Jon L. Bentley, Michael R. Garey, Christos H. Papadimitriou, and Michael I. Shamos.

References

[1] Aho, A. V., J. E. Hopcroft, and J. D. Ullman. *The Design and Analysis of Computer Algorithms.* Addison-Wesley. Reading, Massachusetts (1974).

[2] Garey, M. R., R. L. Graham, D. S. Johnson, and D. E. Knuth. "Complexity Results for Bandwidth Minimization." *SIAM Journal on Applied Mathematics* 34 (1978). pp. 477-495.

[3] Garey, M. R. and D. S. Johnson. *Computers and Intractibility: A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company. San Francisco (1979).

[4] Garey, M. R., D. S. Johnson, and L. Stockmeyer. "Some Simplified NP-Complete Graph Problems." *Theor. Comput. Sci.* 1 (1976). pp. 237-267.

[5] Papadimitriou, C. H. "The NP-Completeness of the Bandwidth Minimization Problem." *Computing* 16 (1976). pp. 263-270.

[6] Shamos, M. I. Private communication (1979).

---

[10] Note that a graph of N vertices has bandwidth ≤ k iff it is isomorphic to a subgraph of a maximal graph of size N and bandwidth k.

First steps toward a new theory of dynamic programming

ARNON ROSENTHAL, PAUL HELMAN

Department of Computer and Communications Sciences,
University of Michigan

Ann Arbor, Michigan 48109

A formal model of implicit enumeration is presented. It applies to dynamic programming and certain other enumeration algorithms. It has both practical and theoretical interest.

INTRODUCTION

There is no universally accepted definition of dynamic programming. Our tentative definition is: Dynamic programming is a divide and conquer procedure in which the subproblems to be generated are known in advance, independent of most of the input data, and in which subproblem solutions are tabulated, so subproblems need be solved only once. The solution space is implicitly searched, using Bellman's principle of optimality ("every subsolution of an optimal solution is optimal for its subproblem") to eliminate suboptimal solutions.

The behavior of a dynamic programming algorithm typically depends on problem size and structure, but not on actual data values. Our model incorporates this restriction and is thus not applicable to quicksort or branch and bound schemes, whose behavior is quite data dependent.

Major goals

1 The model should help nonspecialists to formulate, verify, and explain the recurrence relations used to solve problems by dynamic programming, especially when no natural serial stage structure is present.

2 The model should not be restricted to optimization problems. It should also apply to probability calculations, counting problems, and others which may be solved by implicit enumeration.

3 Optimally efficient enumeration schemes will be studied.

Frequently it is possible to restrict consideration to a small, simple subclass of {all implicit enumeration schemes}.

MODELS

Our theory is motivated by dynamic programming. A key feature of the implicit enumeration associated with dynamic programming is that a comparison enumerates extensions of the partial solutions compared. Thus the elements enumerated are structurally related to the elements being operated on. In contrast, for branch-and-bound, the bounding rule requires little direct connection between the operands and the elimination. The theory of "oracle" algorithms developed by Korte and Haussman has no explicit bounding rule.

Models of dynamic programming For discrete optimization problems solved by exploiting a serial stage structure, models are available. In [KaHe] a discrete optimization problem is formalized as a "discrete decision process". Sequential decision processes" are defined, and modelled as a finite automaton with costs attached. States of the finite automaton are defined by considering an equivalence relation on final segments of the input string.

An ambitious attempt to apply the finite automaton formalism to dynamic programming algorithms is [Brown]. He finds that the formalism must be extended for some problems (e.g. finding optimal binary trees) where the stages are not naturally serial. In addition, the encodings of several problems as finite automata are quite nonintuitive. It seems fair to say that for those problems, dynamic programming recurrences are easier to define and understand without interposing the finite automaton formalism.

[DrLa] takes an informal view, and suggests the "consultant" question : If the system has been operating for a time, and you wanted a consultant to operate it optimally, what information would the consultant need? This information defines the state.

Nonserial problems The limitation of all the above models is that they are basically serial: Given a partial solution, they evaluate the effect of adding on one more decision. It may be much more natural to take partial solutions to two nontrivial subproblems and combine those. Our view will thus subsume the serial view. The consultant question can easily be generalized: if decisions have already been made for several subproblems, the consultant then needs certain information to summarize events in these subsystems. This information defines the states.

There are two benefits of having this wider view. For some problems, there is no natural serial way to break them up. One does not naturally add one node when studying binary trees--one combines subtrees. Second, for problems which do have natural serial structure, one can more easily formalize alternative nonserial algorithms for the problem, and compare them with the serial algorithm.

DEFINITION OF THE ENUMERATION PROCESS

An enumeration problem is specified by a tuple $P=\langle E,S^0,R,r,^0\rangle$ where

E is the set of basic elements

Members of S^0 are the legal stuctures of the system (i.e. legal combinations of element states).

r maps structures into R, the set of structure representatives

^{0} is a communative and associative binary operator on R

The computational task is to compute $\bigcirc_{s \in S^0} r(s)$.

r is extended to map sets of structures into representatives by the rule $r(S1 \cup S2) = r(S1)\,^0\,r(S2)$, for any disjoint sets of structures S1 and S2. Because ^{0} is commutative and associative, and each set includes a finite number of structures, the extended r is well defined.

The key to the efficiency of the enumeration is not the sum operation, but decomposition. A decomposition of an enumeration problem P is a system $P^1,P^2,\ldots$ of smaller enumeration problems, referred to as subproblems. Some axioms will be needed to describe the relationship of the original problem to its subproblems.

Now we present the decomposition rules. We assume that there exists a second partial binary operation, concatenation of structures, denoted *. * is extended to a total function by augmenting the structure set with a "dead" structure denoted d, and augmenting the representatives by adding the representative of d, denoted e. e will act as an identity under ^{0}.

A decomposition of an enumeration problem P into subproblems $P^1, P^2, \ldots$ may now be defined.

Def. $\langle P^1, P^2, \ldots, *, ! \rangle$ is a system of subproblems of the enumeration problem $\langle E^0, S^0, R, r, {}^{o} \rangle$ if each $P^j = \langle E^j, S^{j0}, R^j, r^j, o^j \rangle$ is an enumeration problem such that

i if n1 and n2 are structures of subproblems of the system (called substructures), then so is n1!n2

ii $\cup E^j = E$

iii For any subproblems j and k, $r(S1^j)!r(S2^{k}_{\_}) = r(S1^j * S2^{k}_{\_})$.

iv * distributes over union, and ! distributes over ^{o}. e.g.,

$(r(n)!r(n1))\,^{o}\,(r(n)!r(n2)) = r(n)!\,(r(n1)\,^{o}\,r(n2))$

There are several other axioms governing the behavior of ^{o} and !. They will be discussed at a later date.

The model is an extension of Schnorr's model of monotone computations [Schnorr].

APPLICATIONS

We give a few of the interesting interpretations of the model.

All pairs shortest paths: A structure is a path. A substructure is an (i,j) path, where i and j are fixed for a given subproblem. * concatenates paths, ^{o} compares their lengths, and ! adds their lengths.

Two subtrees T^1 and T^2 are grafted together by creating a new tree with a new root, T^1 as left subtree, T^2 as right subtree.

Alphabetic trees: A structure is an ordered binary tree (i.e., a legal code), and a substructure is a subtree (i.e., a partial code). * grafts subtrees together, ! computes the cost of the union, and ^{o} compares the costs of trees.

Context free language recognition: A structure is a set of labelled trees generating the input string. Substructures are labelled sets of trees generating substrings of the input. * grafts trees together, ! finds sets of nonterminals correspondingto the grafted trees, and and ^{o} is a union operator.

Combinatorical circuits with probabilistic inputs: A structure is a state of the network, and the representative of the structure is the probability of the network being in this state. A substructure is a state of a subnetwork. * combines subnetworks, and ^{0} accumulates probabilities.

OPTIMAL ENUMERATION

The power of implicit enumeration comes from the fact that when the representatives of the sets S1 and S2 are concatenated the result is the representative of a set of up to $|S1| \cdot |S2|$ structures. A natural enumeration strategy is to concatenate representatives of the largest possible sets of structures, i.e., representatives of {all the structures of a subproblem}.

Def An enumeration is a *fully solving enumeration* if concatenation is only performed between representatives of {*all* the structures of the subproblems involved}.

Let C^0 and C denote classes of algorithms, such that $C^0 \subseteq C$. C^0 will be called an *optimal subclass* of C if, for every problem instance, some member of C^0 is optimal among all algorithms in C. We will identify a class of enumeration algoritms for which {enumeration algorithms which fully solve} is an optimal subclass.

Def An enumeration is *consistent* if (r(S1) is concatenated with r(S2)) implies (concatenation is defined between all the structures of S1's subproblem and all the structures of S2's subproblem).

Def An enumeration is *one-one* if (representatives of (S1 and S2) and of (S3 and S4) are concatenated to get structures of the same subproblem) implies (n1!n2 =n3!n4, for all structures ni of Si's subproblem).

Def An enumeration is a *regular enumeration* if it is both consistent and one-one.

The following theorem can be proved:

Theorem {Regular enumerations which fully solve} is an optimal subclass of {regular enumerations}.

FUTURE WORK

We hope to simplify the model to make it more appealing intuitively, and easier for nontheoreticians to use. We are searching for other examples which fit nicely into our model, but not into the standard ones. In previous work, for certain special classes of dynamic programming algorithms, algorithms were given to efficiently mass-produce solutions to an optimization problem in which each element of input data was altered, one at a time [Ros 1,2]. We wish to explore conditions for these schemes to work in the much more general framework considered here. The relative power of finite automaton and function decomposition formalism will be compared. Complexity measures other than (# unions) might be considered. Finally, we would like to identify an optimal subclass of {enumeration algorithms which use storage less than some limit}.

References

[Brown] K. Q. Brown, "Dynamic programming in computer science," Technical report, Department of Computer Science, Carnegie-Mellon University, Feb. 1979.

[DrLa] S. Dreyfuss and A. Law, *The art and theory of dynamic programming*, 1979.

[KaHe] R. M. Karp and M. Held, "Finite -state processes and dynamic programming," SIAM J. Appl. Math 15(3) May 1967 pp. 693-717.

[Ros-1] A. Rosenthal, "Optimal Algorithms for Sensititvity analysis in associative multiplication systems," to appear in *Theoretical Computer Science*.

[Ros-2] A. Rosenthal, Sensitivity analysis in nonserial problems", submitted to J. Math. Anal. and Appl.

[Schnorr] C. P. Schnorr, "Complexity of Monotone Computations' *Theoretical Computer Science*, vol. 2.

CONTINUAL PATTERN REPLICATION*

Preliminary Version

J. IAN MUNRO
Dept. of Computer Science
University of Waterloo
Waterloo, Ontario N2L 3G1

EDWARD L. ROBERTSON
Dept. of Computer Science
Indiana University
Bloomington, IN 47401

ABSTRACT

In this paper propagation of a repeated pattern in an infinite one-dimensional tesselation array is considered. Waksman's [2] technique is shown to work only for certain pattern lengths. This length restriction is shown to apply to any real-time scheme. General linear time techniques, arbitrarily close to real-time, are described, and trade-offs between alphabet and bud size are considered.

I. INTRODUCTION AND NEGATIVE RESULTS

A one-dimensional array of identical finite state machines has been used by Waksman [2] to model the continual replication of n symbols represented by the states of the cells of the array. More precisely, the model is an arbitrarily long one-dimensional array of identical finite state machines. These machines are synchronous and the state of each at time t+1 is a function of its own state and that of its two neighbours at time t. The first cell differs from the others only in that it has no left neighbour. At time t=0 the first f(n) cells (f(n) = n+1 in Waksman's solution) are set to a sequence of internal states. This active sequence of cells is called the generating bud. The remaining cells are in a simple state, Q, the quiescent state. The problem is to define the automata and the initial states of the generating bud in such a manner that, at some future time, each machine will reach a terminal state so that a predetermined sequence of n states (or symbols) will be continually repeated along the array. Another method of viewing the process is that of having a (bounded) sequence of machines which effectively broadcast a message over and over again. One can perhaps think of this in terms of satellite identification. The analogy is, however, not complete and we prefer to think of our problem simply in terms of tesselation automata.

We will use Σ to denote the set of terminal states, which may be used to construct the permanent pattern, and Γ to

---

*Supported by the Natural Science and Engineering Research Council of Canada under Grant A8237.

denote the set of all states, including the quiescent state Q, the terminal states, and any required "working" states. Σ has k states.

Waksman's solution is particularly elegant. For each terminal state σ in Σ, there is a corresponding working state $\underline{\sigma}$. The working states are viewed as integers modulo k. Initially the first cell is in a special state P, and the next n cells are in states related to the desired pattern. All states to the left of P are terminal states. To the right of P there is an active bud of k states, and beyond that only Q's. The pattern in the active bud varies from step to step and is displaced one cell rightward during each step. In particular, at time t+1, cell i enters the state corresponding to the sum of its state and that of its left neighbour, cell i-1, at time t. Arithmetic is modulo k, and, for purposes of arithmetic, Q has "value" 0. A cell whose left neighbour is in state P will enter state P in the next step. The cell in state P takes the state of its right neighbour as a permanent state. Thus a cell in state P trails the k active cells. We see, then, that Γ is of size 2k+2. The size of Γ is easily reduced to 2k by equating Q with the working state corresponding to 0 mod k and making a small change in the second phase to eliminate the state P.

Figure 1 illustrates this approach. Suppose we are to generate "021" continually, so in fact k=n=3. In this case "020" is the appropriate initial form of the generating bud. The first few steps of the automata follow the pattern of Figure 1.

$$\begin{array}{r|cccccccccccc}
t = 0 & P & \underline{0} & \underline{2} & \underline{0} & Q & Q & . & . & . & . & . & . \\
1 & 0 & P & \underline{2} & \underline{2} & \underline{0} & Q & Q & . & . & . & . & . \\
2 & 0 & 2 & P & \underline{1} & \underline{2} & \underline{0} & Q & Q & . & . & . & . \\
3 & 0 & 2 & 1 & P & \underline{0} & \underline{2} & \underline{0} & Q & . & . & . & . \\
4 & 0 & 2 & 1 & 0 & P & \underline{2} & \underline{2} & \underline{0} & . & . & . & .
\end{array}$$

Figure 1

Note that at time 3 the active portion of the array is identical to that at time 0 and one copy of the pattern in "permanent" form has been produced. From this it should be clear that the array behaves as desired.

There is obviously a bit of number theory behind this magic. Indeed Waksman shows that this technique can be used to generate an arbitrary string of length n provided it is over an (output) alphabet of size $k = g_n = \gcd \{ \binom{n}{i} : i=1,..n-1\}$. Furthermore it is shown that the initial states of the generating bud, $\{b_i\}$ are related to the desired output, $\{a_i\}$, by

$$a_i = \sum_{j=1}^{i} \binom{i-1}{j-1} b_j$$

An automaton based method is present for determining $\{b_i\}$ from $\{a_i\}$. It is, however, straightforward to show the simpler form

Theorem 1:

$$b_i = \sum_{j=1}^{i} \binom{i-1}{j-1} (-1)^{i+j} a_j$$

Another observation regarding this basic technique is, however, more disturbing. Ram [1] has shown that if n is of the form p^r, where p is prime and r a natural number, then

$$\gcd \{ \binom{n}{i} : i=1,...,n-1\}$$

is p. Furthermore, if n is not of this form the gcd is 1. Hence we may state:

Theorem 2: The pattern replication scheme outlined above is effective over a non-trivial alphabet if and only if the length of the desired output sequence is of the form p^r for some prime p and natural number r. In this case the output alphabet may be of size p.

Waksman's solution is elegant. More precisely, it is a real time solution (a new symbol is generated at each time step), the alphabet required is only double that of the desired output alphabet, and the generating bud is only as long as the pattern produced. Our goal would be to achieve this type of solution for any pattern and alphabet. However, we can show this is not possible.

Theorem 3: No pattern of (minimum) period n can be generated in real time unless every prime factor of n is at most the alphabet size of the automata.

Proof: Any generation operating in real time must have the property that there is a fixed (or at least non-growing) generating bud between the quiescent cells and the pattern which has been produced. This bud or active segment must move 1 cell to the right at each time step. Hence looking at the bud above, we see that the transition of i^{th} cell from the right end of the bud depends on the previous states of cells 1, 2,..., i from the right. This of course implies that no information is passed to the right relative to the position of the bud.

Note that the transition of the rightmost bud cell depends only on that cell (and two cells in the quiescent state. The state of the lead cell is, then, cyclic of order at most the number of states in the alphabet. Similarly the second cell from the right is cyclic of order the cycle of the lead cell times some number atmost the number of states. Again its minimum cycle contains no factor larger than $|\Gamma|$.

In general, the i^{th} cell from the right is cyclic of order the product of the cycles of its right neighbours and some number atmost $|\Gamma|$. By induction, its minimum period contains no factor larger than $|\Gamma|$. □

We see, then that Waksman's solution is somewhat of an anomaly. We will, however, show techniques for coming reasonably close to the goal outlined above.

II. OTHER APPROACHES

In the remainder of this paper we will adopt the general attitude that the pattern length, n, is very long, particularly in comparison to the desired alphabet size, k. In general, n must be independent of the number of states of the automata. We prefer that the length of the generating bud be linear in n. The basic goal will be to develop pattern replication schemes which operate at a rate close to real time and which have a number of states close to linear in k and a generating bud length close to linear in n. Obviously this goal leaves room for some trade-offs.

With this general goal in mind we outline two basic approaches which are quite different from Waksman's. The simplest and most natural technique is to include $\Sigma \times \Sigma$ as a subset of Γ, and use it to move the pattern along the array in a "tractor tread" fashion. This is illustrated in Figure 2 where the basic pattern to be repeated is "abcc" and so the output alphabet, $\Sigma = \{a,b,c,\}$.

t = 1	c a	c b	b c	a c	Q	Q	.	.	.
2	$\underline{a}$	c b	c c	b c	$\underline{a}$	Q	Q	.	.
3	a	a b	c c	c c	b a	Q	Q	.	.
4	a	$\underline{b}$	a c	c c	c a	$\underline{b}$	Q	Q	.
5	a	b	b c	a c	c a	c b	Q	Q	.
6	a	b	$\underline{c}$	b c	a a	c b	$\underline{c}$	Q	Q
7	a	b	c	c c	b a	a b	c c	Q	Q
8	a	b	c	$\underline{c}$	c a	b b	a c	$\underline{c}$	Q
9	a	b	c	c	c a	c b	b c	a c	Q

Figure 2

In general, let Σ be fixed and $\Gamma = \Sigma \times \Sigma \cup \Sigma \cup \Sigma \cup \{Q\}$, where Σ simply denotes the set of symbols in Σ written with underscores. In Figure 2, states of the form <a,b> are written $\genfrac{}{}{0pt}{}{a}{b}$. Assuming that the pattern length is at least two, the state transition may be defined as in Table 1. The notation a,b => c means that if at time t cell i-1 is in state a and cell i is in state b, then cell i enters state c at time t+1 (this notation is justified since the next state of an automaton does not depend on the current state of its right neighbour). The states u, v, x and y are arbitrary members of Σ and ∅ indicates the left end of the array.

Although this technique can be applied to an arbitrary pattern, $(k+1)^2$ states are required. Furthermore the pattern is produced at "half real-time", so that the i^{th} automaton reaches its final state at time 2i. The latter difficulty is partially rectified, at the expense of a much larger Γ, by encoding several (say p) symbols in the upper track. The basic motion (in the middle of the bud) again has the top track moving right on each move, and the bottom staying still. At

left end		middle	
v, <x,y>	=> $\underline{y}$	<u,v>, <x,y>	=> <u,y>
$\underline{v}$, <x,y>	=> <v,y>	right end	
∅, <x,y>	=> $\underline{y}$	<u,v>, Q	=> $\underline{u}$
∅, $\underline{y}$	=> y	<u,v>, $\underline{y}$	=> <u,y>
v, $\underline{y}$	=> y	u,Q	=> Q

Table 1

the front (right end) of the bud, when a full top track moves onto a formerly quiescent cell, the lead element drops onto the bottom track. Then, each time a "partial" top moves forward, the lead character drops down. A new element is added to the front of the bottom track at each time step until the top track of the leading cell is emptied. At that time one step is missed while a new (full) top track comes forward. Hence, during p of every p+1 steps the front of the bud moves forward. Essentially the same construction is used at the rear of the bud, the top track filling while it moves forward. The example in Figure 3 is more instructive than a formal description of this process. We note that about k^{p+1} states are required.

```
t = 0 | abc def
      |  h   g
    1 |     abc de
      |  h   g   f
    2 |      h  abc  d
      |  h   g   f   e
    3 |          gh abc
      |  h   g   f   e   d
    4 |             fgh abc
      |  h   g   f   e   d
    5 |                 fgh ab
      |  h   g   f   e   d   c
    6 |                  e  fgh  a
      |  h   g   f   e   d   c   b
    7 |                     de  fgh
      |  h   g   f   e   d   c   b   a
```

Figure 3: Generating the string "hgfedcba" at 3/4 speed using the tractor tread method.

There is, of course, the minor problem with the scheme as outlined above. The pattern length must be divisible by p+1.

There are many possible solutions to this difficulty. The easiest is to let the bud contain p+1 concatenated copies of the pattern.

The tractor tread technique is simple because pattern flow and synchronization occur simultaneously. There is, however, quite clearly some wastage caused by the bottom track staying fixed. A technique whereby the information flow is separated from the synchronization requires the data to move forward only at propagation speed. The following "shift and drop" method achieves this end. Again two tracks are employed. The top track again contains the reversal of the desired pattern. This is continuously shifted to the right. As a character on the top moves onto a square on which it belongs, a copy of the character "drops" onto the lower track (it also remains on top). There are two ways to detect this. The first requires marking those automata which will ultimately be a first (leftmost) position of the repeated pattern. When the leading character of the shift-and-drop pattern reaches such a marked automaton, it drops into place. This marking is of course the synchronization alluded to earlier and must be accomplished by preceding the "shift and drop" portion of the bud with a "tractor tread" bud which generates the pattern 10^{n-1}. The "1", of course, indicates the beginning of a pattern replication.

The second means of detecting that a character must drop requires that a previously unoccupied bottom track copy the symbol entering the corresponding top track two steps after the bottom track of its leftmost neighbour changes for empty to occupied. The extra delay is required because the top pattern is reversed. It has the consequence that there may be two partially completed patterns and two characters may drop in the same step. Figure 4 illustrates a simple example of this technique with the synchroniztion problem trivialized.

Observe that the "shift and drop" approach naturally permits propagation in real time, since on the average, one symbol is dropped onto the bottom track at each time step. Unfortunately, Theorem 3 says that the marking of initial pattern positions cannot be done in real time without using $\theta(n)$

t = 1	a	a	b	00	.	.	.	.
	b	a	a	1				
2		a	a	b	10	.	.	.
	b	a	a	1	0			
3			a	a	b	01	.	.
	b	a	b	b	0	0		
4				a	a	b	00	.
	b	a	a	b	a	0	1	
5					a	a	b	10
	b	a	a	b	a	a	1	0

Figure 4

states (for certain values of n). This marking can, however, be done at a rate of p/(p+1) using about 2p states. This leaves the slightly annoying problem of slowing the progress of the "shift and drop" mechanism to a speed of p/(p+1). This may be accomplished by adding a p+1 state counter to each automaton. A technique requiring fewer states splits the count over three adjacent machines. Each machine has a counter with $\lceil p^{1/3} \rceil$ states. A specially marked state keeps the low order portion of the count and increments its counter every time interval. Counters on other machines increment only every time their right neighbour overflows. The counters are reset and the shifting pauses every time that the three adjacent counters represent exactly p. Note that these counters must shift with the rest of the shift drop mechanism, although they are of course discarded when an automaton enters a final state. This multiplies the number of states of an automaton participating in the shift and drop by $2\lceil p^{1/3} \rceil$. An additional 4p states are required preceding the shift-and-drop mechanism to establish the counters (one factor of two is required to save the synchronization bit).

Assuming that n>p, the top row of the binary tractor tread will contain at most one bit which is set, so a p+1 valued counter (zero implies no bit set) is sufficient to record this information. Separate p+1 state cycles at the front and back of the tractor tread record the actual number of bits present. These can be in the end automation, requiring 2(p+1)(p+2) states for the tractor tread, or in immediately adjacent automata, lengthening the bud by two be reducing the number of states required for synchronization to

5p+5.

Combining the tractor tread with the states setting up the counters which cause the shifting to pause at appropriate times (and eliminating some redundancies, the control portion of the bud has 8p+3 states and has length n+2 (unless (p+1)|n, in which case a length of n/(p+1) suffices).

Finally we consider in detail the shift and drop mechanism which actually transmits the pattern information. For the simple shift-and-drop method described above, k(k+2) working states were required, not considering the two distinct initial states resulting from the tractor tread's marking the pattern's placement and not considering the clocking mechanism which causes the shifting to stop every p+1 steps.

For large k, however, it may be considered wasteful to have such a large state set. This can be accomplished by shifting information forward one bit at a time rather than all at once. To be specific, the shift-and-drop uses a binary pattern with length n*log k. The first n-bit segment of this pattern contains the first bit of each state of the repeating pattern, according to some binary encoding of Σ. The second n-bit segement is the second bit of the states, and so on. Recording only as many bits of the final state as necessary while they accummulate, this technique requires 8k working states.

If some shift-and-drop variation requires a bud of length bl and ws working states, then any variation of the above which shifts a uniform amount of information must satisfy

$$\underline{bl}*\log(\underline{ws}/k) \geq n*\log k.$$

This is merely the requirement that enough bits were shifted to record the information in the entire pattern. However, it is possible to package the necessary information in a way which is not rectangular and greatly reduce this product.

Let k' = log k. Visualize the shift-and-drop variation just discussed implemented with k'+1 binary tracks. The top track is used to shift information forward one bit at a time, while the remaining k' bits accumulate information about the final state. However, in the leading part of the bud, few of the bottom k' tracks contain useful information. The variation of shift-and-drop we now propose makes use of this situation, shifting several bits in the early put of the bud.

In particular, the first n-state segment of the shift-and-drop bud will shift $k'/2$ tracks while reserving $k'/2$ tracks for recording final state information. The second n-state segment shifts $k'/4$ bits, adding $k'/4$ bits to the $k'/2$ bits of final state information already present. Similarly the i^{th} segment shifts $k'/(2^i)$ bits and leaves $k'*(1 - 1/2^i)$ bits of the final state in place. Thus log k' segments of length n are required.

The number of states required for this shift-and-drop variation is $2^{k'+1}\lfloor \log k' \rfloor$ which is at worst $4k\lfloor \log \log k \rfloor$ In this case $\underline{bl} \log(\underline{ws}/k) \approx n\lfloor \log \log k \rfloor \lfloor \log \log \log k + 2 \rfloor$.

REFERENCES

[1] Ram, B., Jour. of the Indian Mathematics Club, 1 (Madras, 1909), 39-43.

[2] Waksman, A, "A model of replication," JACM 16, 1 (Jan. 1969), 178-188.

DECOMPOSITIONS IN NONLINEAR AUTONOMOUS SYSTEMS: I — GLOBAL

S.H. MIKHAIL
Dept. of Electrical Engineering
University of Toronto
Toronto, Canada M5S 1A4

ABSTRACT

Certain global structural properties of general nonlinear autonomous systems on C^∞ manifolds are described. The concept of a "decomposition" is introduced, which includes linear A-invariance as a special case. Global closure and invariance properties are given and the general nonlinear disturbance decoupling, and noninteracting control (decoupling) problems, are formulated. For both problems necessary conditions for solvability are derived.

This paper reports on global structural and dynamic properties of systems defined on C^∞ manifolds M, whose dynamics are described by C^∞ vector fields on M. Our objective is to investigate certain control-theoretic problems of the type in [4]. We rely on the work and notation in [1] and [2]. A summary of notation is given at the end of the paper.

Let $D_1, D_2 \subset V(M)$ be two families of vector fields. The join of D_1 and D_2 is $D_1 \vee D_2 := (D_1 \cup D_2)^*$; and the meet of D_1 and D_2 is $D_1 \wedge D_2 := V[P_{D_1}] \cap V[P_{D_2}]$. $P_{D_1 \vee D_2}(x) \supseteq P_{D_i}(x)$, $(i = 1,2)$, and $P_{D_1 \wedge D_2}(x) \subseteq P_{D_1}(x) \cap P_{D_2}(x)$, all $x \in M$. The join and meet operations defined on C^∞ distributions spanned by C^∞ families of vector fields specialize to the operations of addition and intersection of vector spaces in linear systems.

Proposition 1.1:

Let $S \subset V(M)$ be a family of vector fields and $S' := V[P_S] \subset V(M)$. Then $P_S(x) = P_{S'}(x)$, all $x \in M$.

Proposition 1.2:

Let $S_1, S_2 \subset V(M)$ be two families of vector fields. Then, the C^∞ distribution $\Delta_{S_1 \wedge S_2}$ has the integral manifolds property [IMP].

Definition 1.1:

Let $S \subset V(M)$ be a family of vector fields, $X \in V(M)$ a complete vector field and, $\tilde{S} := \{X + X' \mid X' \in S\}$. X is decomposable w.r.t. S if $\Delta^0_{\tilde{S}}(x) = P_S(x)$, all $x \in M$ and, P_S will be called a decomposition of X.

Theorem 1.1:

Let $S_1, S_2 \subset V(M)$ be two families of vector fields, and $X \in V(M)$ a complete vector field such that X is decomposable w.r.t. both, S_1 and S_2. Then X is decomposable w.r.t. $S_1 \cup S_2$.

Lemma 1.1:

Let $S \subset V(M)$ be a family of vector fields, $X \in V(M)$ a complete vector field, and X be decomposable w.r.t. S. If $D' := \{X+X' | X' \in V[P_S]\}$, and $\tilde{S} := \{X+X' | X' \in S\}$, then $\Delta^0_{\tilde{S}}(x) = \Delta^0_{D'}(x)$ for all $x \in M$. (i.e. X is decomposable w.r.t. $V[P_S]$.)

Theorem 1.2:

Let Δ be a C^∞ distribution with the IMP; and $X \in V(M)$ a complete vector field, such that Δ is G_x-invariant. Then X is decomposable w.r.t. $V\ [\Delta]$.

Proposition 1.3:

Let $S \subset V(M)$ be a family of vector fields, $X \in V(M)$ be a complete vector field and $\tilde{S} := \{X+X' | X' \in S\}$. Then $\Delta^0_{\tilde{S}}$ is the smallest G_x-invariant distribution with the IMP that includes Δ_S.

Theorem 1.3:

Let $S_1, S_2 \subset V(M)$ be two families of vector fields, and $X \in V(M)$ a complete vector field, decomposable w.r.t. S_1 and w.r.t. S_2. Then X is decomposable w.r.t. $S_1 \vee S_2$.

Theorem 1.4:

Let $S_1, S_2 \subset V(M)$ be two families of vector fields, and $X \in V(M)$ a complete vector field, decomposable w.r.t. S_1 and w.r.t. S_2. Then X is decomposable w.r.t. $S_1 \wedge S_2$.

If the C^∞ output map $\delta : M \rightarrow Z$ (where Z is a C^∞ manifold of dimension r) is a submersion, then $\text{Ker } d\delta(x)$ is a regular C^∞ distribution with the IMP [3].

Definition 1.2:

Let $D \subset V(M)$ be a family of vector fields, and $X \in V(M)$ a complete vector field such that (a) X is decomposable w.r.t. D and (b) $\Delta_D(x) \subseteq \text{Ker } d\delta(x)$, for all $x \in M$. Thus $P_D(x) \subseteq \text{Ker } d\delta(x)$, $x \in M$, and P_D is a δ-decomposition of X.

Proposition 1.4:

Let $X \in V(M)$ be a complete vector field, and $\delta : M \rightarrow Z$ a C^∞ submersion. There exist a supremal distribution Δ^* with the IMP that is a δ-decomposition of X (i.e. for any $D \subset V(M)$ such that P_D is a δ-decomposition of X, $P_D(x) \subseteq \Delta^*(x)$, all $x \in M$).

Definition 1.3:

Let $S_1, S_2, \ldots, S_k \subset V(M)$ be k families of vector fields and $X \in V(M)$ a complete vector field decomposable w.r.t. $S_i (i \in \underline{k})$. If $\tilde{S} := \{X+X' \mid X' \in \bigcup_{i=1}^{k} S_i\}$, P_{S_i} are said to be independent decompositions of X (ID) if P_{S_i} is $G_{\tilde{S}}$-invariant, $(i \in \underline{k})$.

Theorem 1.5:

Let $S_1,\ldots,S_k \subset V(M)$ be k families of vector fields, $X \in V(M)$ a complete vector field, such that P_{S_i} are ID's of X. Then

(i) $P_{S_{i_1} \cup S_{i_2} \cup \ldots \cup S_{i_j} \ldots \cup S_{i_k}}(x) = P_{S_{i_1}}(x) + P_{S_{i_2}}(x) + \ldots + P_{S_{i_j}}(x) + \ldots + P_{S_{i_k}}(x)$, $(i_j \in \underline{k},\ j \in \underline{k})$.

(ii) $P_{S_{i_1} \cup S_{i_2} \cup \ldots \cup S_{i_k}}(x)$ are $G_{\tilde{S}}$-invariant, where $\tilde{S} := \{X+X' \mid X' \in \bigcup_{j=1}^{k} S_j\}$.

Proposition 1.5:

Let $S_1, S_2 \subset V(M)$ be two families of vector fields, $X \in V(M)$ a complete vector field such that P_{S_1}, P_{S_2} are ID's of X. Furthermore if

$G_i := \{g \in C^\infty(M) \mid (dg)Y(x) = 0,\ x \in M,\ Y \in V[P_{S_i}]\}$, $(i = 1,2)$,

$R_1 := \{\Sigma g_\alpha X^\alpha + Y \mid g_\alpha \in G_2,\ X^\alpha \in S_1,\ Y \in S_1 \wedge S_2\}$ and

$R_2 := \{\Sigma g_\beta X^\beta + Y \mid g_\beta \in G_1,\ X^\beta \in S_2,\ Y \in S_1 \wedge S_2\}$. Then,

(i) P_{R_1}, P_{R_2} are ID's of X, and

(ii) If $\mathcal{X} := \{X + \Sigma g_\alpha X^\alpha + g_\beta X^\beta + Y \mid g_\alpha \in G_1, g_\beta \in G_2, X^\alpha \in S_2, X^\beta \in S_1, Y \in S_1 \wedge S_2\} \subset V(M)$, then

$P_{R_i}(x) = P_{S_i}(x)$ are $G_{\mathcal{X}}$-invariant $(i = 1,2)$

(iii) P_{R_1}, P_{R_2} are decompositions of X.

Definition 1.4:

A control system Σ is a quintuple $(M,\mathcal{U},Z,X,\delta)$, $M,\mathcal{U},Z$ are n, m and r dimensional C^∞ manifolds, $\mathcal{U}$ is called the control manifold, $X: M \times \mathcal{U} \to TM$ is a C^∞ map such that for every $u \in U$, $X(\cdot,u) \in V(M)$. A C^∞ map $\phi: M \to U$ such that $X(\cdot,\phi(\cdot)) \in V(M)$ is complete is called a static state-feedback, and a C^∞ map $\Gamma: M \times U \to TU$ such that $(X,\Gamma) \in V(M \times \mathcal{U})$ is complete is called a dynamic state-feedback. Let $\overline{\Omega} := \{C^\infty \text{ maps } \phi: M \to \mathcal{U} \mid X(\cdot,\phi(\cdot)) \in V(M) \text{ is complete}\}$, and $X_\Omega := \{X(\cdot,\phi(\cdot)) \in V(M) \mid \Omega \subset \overline{\Omega}\}$. Then $\overset{0}{\Delta}_{X_\Omega}$ is called a decomposition of Σ and is obtained by switching from one feedback to another in $\overline{\Omega}$.

Proposition 1.6:

Let $\Omega_1, \Omega_2 \subset \overline{\Omega}$ and let the C^∞ map $\phi: M \to \mathcal{U}$ belong to $\Omega_1 \cap \Omega_2$, and $S_i := V[\overset{0}{\Delta}_{X_{\Omega_i}}]$ $(i = 1,2)$. The $P_{S_1 \cup S_2}(x) = \overset{0}{\Delta}_{X_{\Omega_1 \cup \Omega_2}}(x)$ $(x \in M)$ is also a decomposition of Σ, and is a decomposition of $X(\cdot,\phi(\cdot))$.

Let $\Delta:M\times U\rightarrow TM\times TU$ be a C^{∞} distribution with the IMP, $\Gamma\{\Delta\}:=$ {a family of C^{∞} maps $\Gamma:M\times U\rightarrow TU \mid X_{\Gamma}:=(X(x,u),\Gamma(x,u))\in V(M\times U)$ is complete and Δ is $G_{X_{\Gamma}}$-invariant}, $\Gamma_c\{\Delta\}:=V_{\Delta}\wedge V_{TU}$ and $\mathcal{X}\{\Gamma\{\Delta\}\}:=\{(X,\Gamma)\in V(M\times U)\mid \Gamma\in\Gamma\{\Delta\}\}$.

Definition 1.5:

For every $Y\in\mathcal{X}\{\Gamma\{\Delta\}\}$, and $Y':=\{Y+X'\mid X'\in\Gamma_c\{\Delta\}\}$, $\Delta^{0}_{Y'}$ is called a dynamic decomposition of Σ w.r.t. Y. (DDΣ,Y).

Lemma 1.2:

Let $Y\in\mathcal{X}\{\Gamma\{\Delta\}\}$, and $Y':=\{Y+X'\mid X'\in\Gamma\ \{\Delta\}\}$. Define the sequence

$S_0 := \Gamma_c\{\Delta\}$

$T_{\mu+1}(x,u) := \{\, dg\cdot\tilde{X}\cdot g^{-1},\ g = G_Y,\ \tilde{X}\in S_{\mu}\,\}$

$R_{\mu+1}(x,u) := \{\, dh\cdot Z\cdot h^{-1},\ h\in G_{V[T_{\mu+1}]},\ Z\in V[T_{\mu+1}]\}$

$S_{\mu+1} := V[R_{\mu+1}]\in V(M\times \mathcal{U})$, $\mu = 0,1,2,3,\ldots$

Then, $R_{\mu+1}(x,u)\subseteq\Delta^{0}_{Y'}(x,u)$, $(x,u)\in M\times\mathcal{U}$, all μ; and if for some $\mu = k$, $R_k(x,u) = R_{k+1}(x,u)$, all $(x,u)\in M\times\mathcal{U}$, then $R_k(x,u) = \Delta^{0}_{Y'}(x,u)\subseteq\Delta(x,u)$, $(x,u)\in M\times\mathcal{U}$.

In Lemma 1.2, for each $Y\in\mathcal{X}\{\Gamma\{\Delta\}\}$ we get a different $\Delta^{0}_{Y'}$, but they are all included in Δ.

Definition 1.6:

The smallest C^{∞} distribution with the IMP that is $G_{\mathcal{X}\{\Gamma\{\Delta\}\}}$-invariant and includes all Δ^{0}_{Y}, $Y\in\mathcal{X}\{\Gamma\{\Delta\}\}$ is called a generalized dynamic decomposition of Σ w.r.t. $\mathcal{X}\{\Gamma\{\Delta\}\}$ (GDDΣ,$\mathcal{X}\{\Gamma\{\Delta\}\}$); denoted by $\Delta^{c}_{\mathcal{X}\{\Gamma\{\Delta\}\}}$.

Lemma 1.3:

Define the sequence, $S_0 := \Gamma_c\{\Delta\}$,

$T_{\mu+1}(x,u) := \{dg\cdot\tilde{X}\cdot g^{-1},\ g\in G_{\mathcal{X}\{\Gamma\{\Delta\}\}},\ \tilde{X}\in S_{\mu}\}$

$R_{\mu+1}(x,u) := \{dh\cdot Z\cdot h^{-1},\ h\in G_{V[T_{\mu+1}]},\ Z\in V[T_{\mu+1}]\}$

$S_{\mu+1} := V[R_{\mu+1}]$, $\mu = 0,1,2,3\ldots$

Then, $R_{\mu}(x,u)\subseteq\Delta^{c}_{\mathcal{X}\{\Gamma\{\Delta\}\}}(x,u)$, all μ, all $(x,u)\in M\times\mathcal{U}$. For some $\mu = k$, $R_k(x,u) = R_{k+1}(x,u) = \Delta^{c}_{\mathcal{X}\{\Gamma\{\Delta\}\}}(x,u)\subseteq\Delta(x,u)$, all $(x,u)\in M\times\mathcal{U}$.

Lemma 1.4:

Let $\Gamma_1,\Gamma_2,\ldots,\Gamma_{\ell},\ldots\in\Gamma\{\Delta\}$, $\mathcal{F}:=\{\Sigma\lambda_i\Gamma_i\mid\Gamma_i\in\Gamma\{\Delta\},\ \lambda_i\in\mathbb{R}$, and $\Sigma\lambda_i = 0\}$, and $G\{\Delta\}:=\{f\in C^{\infty}(M\times\mathcal{U})\mid dfX'(x,u) = 0,\ X'\in V[\Delta],\ (x,u)\in M\times\mathcal{U}\}$. If $\tilde{X}:=\{Y+\Sigma f_iF_i\mid Y\in\mathcal{X}\{\Gamma\{\Delta\}\},\ F_i\in\mathcal{F}$ and $f_i\in G\{\Delta\}\}$, then for any $\tilde{Y}\in\tilde{X}$ that is complete, with $\tilde{Y}':=\{\tilde{Y}+Y'\mid Y'\in\Gamma_c\{\Delta\}\}$, $\Delta^{0}_{\tilde{Y}'}(x,u)\subseteq\Delta^{c}_{\mathcal{X}\{\Gamma\{\Delta\}\}}$ and $\Delta^{c}_{\mathcal{X}\{\Gamma\{\Delta\}\}}(x,u) = \Delta^{c}_{\tilde{X}}(x,u)$.

Furthermore Δ is $G_{\tilde{X}}$-invariant.

Some Basic Systemic Problems:

(i) General Disturbance Decoupling Problem (GDDP)

A dynamic system with disturbance (DSD) is a hexatuplet $(M,\mathcal{U},Z,Q,X,\delta)$ where $M,\mathcal{U},Z$ and Q are n,m,r and s dimensional C^{∞} manifolds, where Q is called a disturbance-parameter manifold. $X:M\times\mathcal{U}\times Q\rightarrow TM$ is a C^{∞} mapping, such that for every $(u,q)\in\mathcal{U}\times Q$, $X(x,u,q)\in V(M)$, and $\delta:M\rightarrow Z$ is the output map (usually assumed to be regular).

Let $\pi\subset Q$ be a set of disturbance parameters. We say that GDDP is solvable on M w.r.t. variations of q in π if there exists a map $\phi:M\times\pi\rightarrow\mathcal{U}$ such that for each $q\in\pi$, $\phi:M\times q\rightarrow U$ is a C^{∞} map, and

(1) $X(x;\phi(x,q);q)\in V(M)$ is complete,

(2) If $X^{\pi}_{\phi} := \{X(x,\phi(x,q),q)\,|\,q\in\pi\}$, then $\overset{0}{\Delta}_{X^{\pi}_{\phi}}(x)\subseteq \operatorname{Ker} d\delta(x)$ all $x\in M$.

This implies that along the trajectories of $X\in X^{\pi}_{\phi}$, $d\delta X = \dot{\delta}\in C^{\infty}(M)$ is independent of variations of q in π.

Similarly one says that GDDP is solvable on M×U w.r.t. variations of q in π if there exists a map $\Gamma:M\times\mathcal{U}\times\pi\rightarrow T\mathcal{U}$ such that for each $q\in\pi$, $\Gamma:M\times\mathcal{U}\times q\rightarrow T\mathcal{U}$ is a C^{∞} map, and

(1) $(X(x,u,q),\ \Gamma(x,u,q))\in V(M\times\mathcal{U})$ is complete,

(2) If $X^{\pi}_{(x,\Gamma)} := \{(x,\Gamma)\,|\,q\in\pi\}$, then $\overset{0}{\Delta}_{X^{\pi}_{(x,\Gamma)}}(x,u)\subseteq(\operatorname{Ker} d\delta^{(x)},\ T\mathcal{U})$.

all $(x,u)\in M\times U$.

One could view the maps $\phi:M\times\pi\rightarrow\mathcal{U}$ (or $\Gamma:M\times\mathcal{U}\times\pi\rightarrow TU$) as a set of parameterized feedback maps. If there exists a single C^{∞} map $\bar{\phi}:M\times q_0\rightarrow\mathcal{U}$ such that for $X^{\pi}_{\bar{\phi}} := \{X(x,\bar{\phi}(x,q_0),q)\,|\,q\in\pi\}$, $\overset{0}{\Delta}_{X^{\pi}_{\phi}}(x)\subseteq\operatorname{Ker} d\delta$, all x M, then GDDP is said to be ideally solvable on M, w.r.t. variations of q in π.

(ii) Generalized Decoupling Problem (GDP)

Let Σ be a control system (M,U,Z,X,δ) where $Z = \otimes Z_i$ $(i\in \underline{k})$ and each Z_i is a C^{∞} manifold of dim r_i, and $\delta := (\delta_1,\delta_2,\ldots,\delta_k)$, $\delta_i:M\rightarrow Z_i$ are maps, such that all maps $(\delta_{\alpha_1},\delta_{\alpha_2},\ldots,\delta_{\alpha_\ell}):M\rightarrow Z_{\alpha_1}\times Z_{\alpha_2},\ldots\times Z_{\alpha_\ell}$ are C^{∞} submersions $(\alpha_i\in\underline{k},\ i\in\underline{k})$.

A system Σ is said to be statically decoupled if there exists a C^{∞} feedback map $\phi_0:M\rightarrow\mathcal{U}$ such that $X(x,\phi_0(x))\in V(M)$ is complete and there are K sets of C^{∞} feedback maps $\Phi_i := \{\phi^j_i:M\rightarrow\mathcal{U}\,|\,j\in\underline{\sigma}_i\}$ with $\phi_0\in\Phi_i$, $(i\in\underline{k})$ and $X_{\Phi_i} := \{X(x,\phi^j_i(x))\,|\,\phi^j_i\in\Phi_i\}$ with the following properties.

(1) $V[\overset{0}{\Delta}_{X_{\Phi_i}}]$ are independent dynamic decompositions w.r.t. $X(x,{}_0(x))$

[i.e. if $\Phi := \bigcup_{i=1}^{k}\Phi_i$, $X_{\Phi} := \{X(x,\phi)\,|\,\phi\in\Phi\}$, $\overset{0}{\Delta}_{X_{\Phi_i}}$ is $G_{X_{\Phi}}$-invariant].

(2) Each set of static feedback maps Φ_i affects only the output δ_i only. Hence, if $\bar{\delta}_i := (\delta_1, \delta_2, \ldots, \delta_{i-1}, \delta_{i+1}, \ldots, \delta_k)$ then $\Delta^0_{X_{\Phi_i}}(x) \subseteq \text{Ker } d\bar{\delta}_i(x) = \bigcap_{j \neq i} \text{Ker } d\delta_j(x)$, $x \in M$.

(3) Allowing for time reversal, the i^{th} output is completely dynamically controllable, via switching between controls of the set Φ_i. This is equivalent to the transversality conditions.

$$\text{Ker } d\delta_i(x) + \Delta^0_{X_{\Phi_i}}(x) = TM_x, \quad \text{all } x \in M, \quad (i \in \underline{k}).$$

We have already mentioned some properties of independent decompositions in Theorem 1.5 which apply here as well.

If time reversal is not allowed, which is usually the case for almost all practical systems, then we have an additional sufficient condition that guarantees that the outputs are dynamically controllable given by $\Delta^0_{\Phi_i}(x) = J^0(X_{\Phi_i})(x)$ $x \in M$, $(i \in \underline{k})$. This is automatically satisfied in the cases of analytic and linear systems. Condition (iii) however remains necessary in all cases.

We shall say that a system Σ is dynamically decoupled if there exists a C^∞ feedback map $\Gamma_0 : M \times \mathcal{U} \to T\mathcal{U}$ such that $(X(x,u), \Gamma_0(x,u)) \in V(M \times \mathcal{U})$ is complete and there are k sets of dynamic feedback maps $\{\Gamma_{ci}\} := \{\Gamma^j_{ci} : M \times \mathcal{U} \to T\mathcal{U} \mid \underline{j \in \sigma_j}\}$ such that $\Gamma_0 \in \{\Gamma_{ci}\}$ $(i \in \underline{k})$ and $X_{\{\Gamma_{ci}\}} := \left\{(X, \Gamma^j_{ci}) \in V(M \times \mathcal{U}) \mid \Gamma^j_{ci} \in \{\Gamma_{ci}\}\right\}$ with the following properties,

(i) $V[\Delta^0_{X_{\{\Gamma_{ci}\}}}]$ $(i \in \underline{k})$ are independent dynamic decompositions w.r.t. (X, Γ_0) in $M \times \mathcal{U}$.

(ii) Each set of dynamic feedback maps $\{\Gamma_{ci}\}$ affects the i^{th} output only. Hence if $\bar{\delta}_i := (\delta_1, \delta_2, \ldots, \delta_{i-1}, \delta_{i+1}, \ldots, \delta_k)$ is the map $\bar{\delta}_i : M \times \mathcal{U} \to Z_1 \times Z_2 \times \ldots \times Z_{i-1} \times Z_{i+1} \times \ldots \times Z_k$, then

$$\Delta^0_{X_{\{\Gamma_{ci}\}}}(x,u) \subseteq \text{Ker } d\bar{\delta}_i(x,u) \subseteq \text{Ker } d\bar{\delta}_i(x) \times T\mathcal{U} \quad (i \in \underline{k}).$$

(iii) Allowing for time reversal the i^{th} output is completely dynamically controllable via switching between controls of the set $\{\Gamma_{ci}\}$. Hence we have the transversality condition

$$\text{Ker } d\delta_i(x,u) + \Delta^0_{X_{\{\Gamma_{ci}\}}}(x,u) = TM \times T\mathcal{U}_{(x,u)}$$

for all $(x,u) \in M \times \mathcal{U}$, and $(i \in \underline{k})$.

Other results of less importance are not given here and are based on the above theory. Proofs have been eliminated due to their excessive length and will be available in [5] and [6].

Discussion:

Consider the linear system $\dot{x} = Ax$, $x \in \mathbb{R}^n$ and $Ax \in V(\mathbb{R}^n)$. Let $\mathcal{V}$ be a vector subspace of $T\mathbb{R}^n (\approx \mathbb{R}^n)$. $\mathcal{V}$ is said to be A-invariant if $A\mathcal{V} \subset \mathcal{V}$. Let $\mathcal{V}_1, \mathcal{V}_2$ be two A-invariant vector subspace, i.e. $A\mathcal{V}_i \subset \mathcal{V}_i$ $(i = 1,2)$, then $A(\mathcal{V}_1 + \mathcal{V}_2) \subset (\mathcal{V}_1 + \mathcal{V}_2)$ and $A(\mathcal{V}_1 \cap \mathcal{V}_2) \subset \mathcal{V}_1 \cap \mathcal{V}_2$. The subspaces $\mathcal{V}_i$ are actually distributions of constant dimension, i.e. $\mathcal{V}_i : \mathbb{R}^n \to T\mathbb{R}^n$. Theorem 1.3 is a generalization of the fact that $A(\mathcal{V}_1 + \mathcal{V}_2) \subset (\mathcal{V}_1 + \mathcal{V}_2)$, and Theorem 1.4 is a generalization of $A(\mathcal{V}_1 \cap \mathcal{V}_2) \subset \mathcal{V}_1 \cap \mathcal{V}_2$.

Consider now $\bar{S} \subset T\mathbb{R}^n$ a vector subspace, then define the sequence of vector subspaces $\mathcal{S}_{\mu+1} := A\mathcal{S}_\mu + \mathcal{S}_\mu$ where $\mathcal{S}_0 = \bar{S}$, $\mu = 0,1,2,3,\ldots$. Then, at some $\mu = k$, $\mathcal{S}_k = \mathcal{S}_{k+1}$ and $A\mathcal{S}_k \subset \mathcal{S}_k$. $\mathcal{S}_k$ here corresponds to $\Delta^0_{\tilde{s}}$ in Proposition 1.5, and is indeed the smallest A-invariant subspace that includes $\bar{S}$ as in Proposition 1.3.

Consider now the linear dynamic system $\Sigma := (\mathbb{R}^n, \mathbb{R}^m, (Ax+Bu), Dx)$, where $\dot{x} = (Ax+Bu)$, $z = Dx$. Let $\bar{\Omega} :=$ {a set of linear feedback maps $F : \mathbb{R}^n \to \mathbb{R}^m$}, and $X_{\bar{\Omega}} := \{(A+BF_i)x \in V(\mathbb{R}^n) \mid F_i \in \bar{\Omega}\}$, then the subspace $\mathcal{V}' := J^0\{X_{\bar{\Omega}}\}$ corresponds to the linear subspace which is the counter part of $\Delta^0_{X_{\bar{\Omega}}}$, in Definition 1.4, and $(A+BF_i)\mathcal{V}' \subset \mathcal{V}'$, all $F_i \in \bar{\Omega}$.

Let $\mathcal{V}$ be a linear subspace, $F : \mathbb{R}^n \to \mathbb{R}^m$ a linear feedback map such that $(A+BF)\mathcal{V} \subset \mathcal{V}$, and $Y = (A+BF)x \in V(\mathbb{R}^n)$ and $\Gamma_c\{\mathcal{V}\} = \mathcal{B} \cap \mathcal{V}$. Then Δ^0_Y in Definition 1.5 corresponds to the largest controllability subspace $R \subset \mathcal{V}$ [4]. In linear systems it has been shown that if $F' : \mathbb{R}^n \to \mathbb{R}^n$ with the property $(A+BF')\mathcal{V} \subset \mathcal{V}$, and $Y' := (A+BF')x \in V(\mathbb{R}^n)$, then $\Delta^0_Y = \Delta^0_{Y'} = R$ [4]. In nonlinear systems however that need not be true, and switching from one feedback to the other will in general give us more flexibility and extend the dimension of our reachable sets. Hence the generalized dynamic decomposition of Definition 1.6 is a generalization of linear controllability subspaces.

List of Symbols

$V[\Delta]$	the set of all C^∞ vector fields defined on M and span Δ.
$V(M)$	the set of all C^∞ vector fields defined on M.
P_D	the smallest G_D-invariant distribution that includes Δ_D where $D \subset V(M)$ is a family of vector fields, including the zero field.
Δ_D	the C^∞ distribution spanned by D.
$(D)^*$	the smallest Lie subalgebra of V(M) that includes the family $D \subset V(M)$.
Δ^0_D	$:= \{\Sigma \lambda_i dg_i X^i g_i^{-1}(x),\ g \in G_D,\ X^i \in D,\ \lambda_i \in \mathbb{R},\ \sum_i \lambda_i = 0\}$.
G_X	the group of local diffeomorphisms X_t generated by $X \in V(M)$.

G_D the smallest group of local diffeomorphisms, that contain the union of G_X, $X \in D$.

References

1. H.J. Sussmann, "Orbits of Families of Vector Fields and Integrability of Distributions", Trans. Amer. Math. Soc., 180, (1973), pp. 171-188.

2. N. Kalouptsidis, "Accessibility and Stability Theory of Nonlinear Control Theory", Ph.D. Thesis, 1977, Washington University, St. Louis, Missouri.

3. Brickell and Clark, Differentiable Manifolds, Van Nostrand, 1970.

4. W.M. Wonham, Linear Multivariable Control: A Geometric Approach, Springer-Verlag, N.Y., second edition, 1979.

5. S.H. Mikhail, W.M. Wonham, "Global Decompositions in Nonlinear Autonomous Systems", Systems Control Group Report No. 7908, Dept. of Electrical Engineering, University of Toronto, (in preparation).

6. S.H. Mikhail, Ph.D. Thesis (in preparation), Dept. of Electrical Engineering, University of Toronto.

DECOMPOSITIONS IN NONLINEAR AUTONOMOUS SYSTEMS: II — LOCAL

S.H. MIKHAIL
Dept. of Electrical Engineering
University of Toronto
Toronto, Canada M5S 1A4

ABSTRACT

Sufficient conditions based on the implicit function theorem establishing and constructing local C^∞ feedback maps such that a system $\dot{x} = F(x,u)$ is locally decomposable are given. The concepts of locally sufficiently amenable and locally weakly controllable and distributions are introduced, and are equivalent to (A,B)-invariant and controllability subspaces in linear systems. Closure conditions and properties of the LSA and LWC properties are given. Local necessary and sufficient conditions for solving the disturbance decoupling problem $\dot{x} = F(x,u,q)$, $z = \delta(x)$ locally, and the noninteracting controls (decoupling) problem, $\dot{x} = F(x,u)$, $z_i = \delta_i(x)$ $(i \in \underline{k})$ are given.

In Part I, global structural properties of systems on C^∞ manifolds were studied. In this paper we investigate the problem of establishing a priori the possibility of locally altering the system vector field $F(x,u)$ by a local C^∞ static feedback map $\phi: U \to W$ where $U \subset X$ and $W \subset \mathcal{U}$ are open sets (or a C^∞ dynamic feedback map $\Gamma: V \to W'$, $V \subset X \times \mathcal{U}$, $W' \subset T\mathcal{U}$) such that a given C^∞-involutive distribution (ID), $\Delta: X \to TX$ (or $\Delta': X \times \mathcal{U} \to TX \times T\mathcal{U}$) of constant dimension on U (or V), is $G_{F(\cdot,\phi)}$-invariant (or $G_{X,\Gamma}$-invariant) for all $x \in U$ (or $(x,u) \in V$).

2.1 Local Decomposability

Definition 2.1:

A dynamic system Σ is said to be statically (dynamically) locally decomposable on an open set $U \subset X$ $(V \subset X \times \mathcal{U})$ w.r.t. a C^∞ (ID) Δ if

(i) $\dim \Delta(x) = \text{constant} = c$, $x \in U$, $(\dim \Delta(x,u) = c$, $(x,u) \in V)$, and

(ii) There exists a cover for U (or V) by a countable family of open sets $\{U_\alpha\}$ $(\{V_\alpha\})$, and for every set U_α, there exist C^∞ maps $\phi_\alpha: U_\alpha \to \mathcal{U}$ $(\Gamma_\alpha: V_\alpha \to T\mathcal{U})$ such that Δ is G_{F_ϕ}-invariant $(G_{F,\Gamma}$-invariant) for all $x \in U_\alpha$ $((x,u) \in V_\alpha)$.

A point $x_0 \in X$ is called a regular point of Δ if there exists a neighbourhood (nbhd.) U of x_0, such that $\dim \Delta(x') = \text{constant}$, $x' \in U$.

Definition 2.2:

A C^∞ ID Δ is locally amenable on $U \subset X$ (or $V \subset X \times \mathcal{U}$) if

(a) Σ is statically (or dynamically) decomposable on U (or V),

(b) $\Delta(x) \subseteq \operatorname{Ker} d\ (x)$, all $x \in U$ (or $\Delta(x,u) \subseteq \operatorname{Ker} d\delta$, $(x,u) \in V$).

Definition 2.3:

Let $[M]$ be an $r \times m$ matrix whose entries are C^∞ functions in $\mathbb{R}^s$. $[M]$ is said to have a "generic rank" K (rK_G) on an open set $U \subset \mathbb{R}^s$ if the set $W := \{x \epsilon U \mid \text{rank } [M]_x < K\}$ is a closed set with an empty interior.

The following theorem gives sufficient conditions for establishing the existence of local C^∞ static feedback maps, such that Σ is locally decomposable w.r.t. Δ, on some open set in $\mathcal{X}$

Theorem 2.1:

Let $\Delta : \mathcal{X} \to T\mathcal{X}$ be a C^∞-involutive distribution (C^∞-ID). If there exists an open nbhd. Ω of (x_0, u_0) in $\mathcal{X} \times \mathcal{U}$ such that

(i) For $\rho := \{x \epsilon \mathcal{X} \mid \exists u \epsilon \mathcal{U},\ (x,u) \epsilon \Omega_0\}$ $\qquad \dim \Delta(x) = \sigma = \text{constant} \quad x \epsilon \rho$ (2.1a)

Then there exists a countable cover of open sets $\{U_\alpha\}$ for ρ and local C^∞ submersions $f_\alpha : U_\alpha \to \mathbb{R}^{n-\sigma}$ such that $(df_\alpha) V_\Delta(x) = 0 \quad x \epsilon U_\alpha$, all α

(ii) The matrices

$$G_\alpha := \begin{bmatrix} d_1[df_\alpha F(x,u)] & d_2[df_\alpha F(x,u)] \\ d_1 f_\alpha(x) & 0 \end{bmatrix} \tag{2.1b}$$

have a generic rank $\ell + (n-\sigma)$ for $(x,u) \epsilon \Omega \cap (U_\alpha \times \mathcal{U})$, all α.

(iii) For every local C^∞ submersion $\bar{f}_\alpha : \tilde{U}_\alpha \to \mathbb{R}^\ell$ (where $\bar{f}_\alpha = \bar{\pi}_\alpha . f_\alpha$ where $\bar{\pi}_\alpha : \bar{V}_\alpha \to \bar{W}_\alpha$, $\bar{V}_\alpha \subset \mathbb{R}^{n-\sigma}$, $\bar{W}_\alpha \subset \mathbb{R}^\ell$ is a local C^∞ submersion) and $\bar{\phi}_\alpha = d\bar{f}_\alpha F(x,u)$ such that

$$\text{rank } G_\alpha(x.u) = \text{rank} \begin{bmatrix} d_1\bar{\phi}_\alpha & d_2\bar{\phi}_\alpha \\ d_1 f_\alpha & 0 \end{bmatrix} = \text{rank } df_\alpha + \text{rank } d_2\bar{\phi}_\alpha$$

for all $(x \epsilon u) \epsilon \Omega \cap ([U_\alpha \cap \tilde{U}_\alpha] \times \mathcal{U})$ we have that,

(a) For all $X \epsilon V_\Delta \subset V(\rho)$, the matrices

$$R_\alpha[X] := [d_2\bar{\phi}_\alpha]^T([d_2\bar{\phi}_\alpha][d_2\bar{\phi}_\alpha]^T)^{-1}[d_1\bar{\phi}_\alpha][X] \text{ exist, and} \tag{2.1c}$$

(b) $T_\alpha := [d_2\bar{\phi}_\alpha]([d_2\bar{\phi}_\alpha][d_2\bar{\phi}_\alpha]^T)^{-1}[d_2\bar{\phi}_\alpha]$ exist (2.1d)

for all $(x,u) \epsilon \Omega \cap ([U_\alpha \cap \tilde{U}_\alpha] \times \mathcal{U})$

Then there exist open sets V_α, W_α nbhds. of x_0' and u_0' in $\mathcal{X}$ and $\mathcal{U}$ respectively such that $V_\alpha \times W_\alpha \epsilon \Omega$, and C^∞ maps $\phi_\alpha : V_\alpha \to W_\alpha$ such that

$$d[df_\alpha(x) F(x_1 \phi_\alpha(x))] V_\Delta(x) = 0 \quad x \epsilon V_\alpha \tag{2.1e}$$

Furthermore there exists a unique C^∞-involutive distribution $\bar{\Delta}^1 : \Omega \to T\Omega$ called the first associated distribution of Δ on Ω (1st ADΔ,Ω) such that

$$df_\alpha V_{\bar{\Delta}1}(x,u) = 0, \quad d[df_\alpha F(x,u)] V_{\bar{\Delta}1}(x,u) = 0, \quad \dim \bar{\Delta}_1 = n+m-\sigma-\ell$$

and there exist open sets $W_\alpha' \subset \Omega \cap (U_\alpha \times \mathcal{U})$ and C^∞ maps $\Gamma_\alpha : W_\alpha' \to T\mathcal{U}$ such that

$$d[d_1[df_\alpha F(x,u)]F(x,u) + d_2[df_\alpha F(x,u)]\Gamma_\alpha(x,u)]V_{\bar{\Delta}^1}(x,u) = 0 \quad (x,u)\in W'_\alpha . \tag{2.1f}$$

Remarks:

(i) The local C^∞ID $\bar{\Delta}_c:\Omega \to T\mathcal{U}$ defined by $V[\bar{\Delta}_c] := V[\bar{\Delta}^1]\wedge V[T\mathcal{U}]$ on Ω is called the control distribution of Δ and is of constant dimension $m-\ell$ on Ω

(ii) Let $\{\Gamma_\alpha\} :=$ {all local C^∞ maps $\Gamma_\alpha : W'_\alpha \to T\mathcal{U}$ satisfying (2.1f)} and $\{\overline{\Gamma_\alpha}\} := \{\Sigma\lambda_i\Gamma_{\alpha i} \,|\, \lambda_1 \in \mathbb{R}_1\ \Gamma_{\alpha i}\in\{\Gamma_\alpha\}$ and $\Sigma\lambda_i = 0\}$, then span $\{\overline{\Gamma_\alpha}\} = T\mathcal{U}$ on $(x,u)\in W_\alpha$. Furthermore on W_α, there exist m+1 vector fields $\Gamma_{\alpha,0},\ldots,\Gamma_{\alpha,m}$ that span $\{\Gamma_\alpha\}$ and $\Gamma_{\alpha 1},\ldots,\Gamma_{\alpha,m}$ spans$\{\overline{\Gamma_\alpha}\}$.

Definition 2.4:

A C^∞-involutive distribution $\Delta:\mathcal{X}\to T\mathcal{X}$ is said to be locally sufficiently amenable on a nbhd. Ω of the operating point (x_0,u_0), LSA(Ω), if

(i) It satisfies (2.1a,b,c,d), and

(ii) $\Delta(x) \subseteq \operatorname{Ker} d\delta(x), \quad x\in\rho$.

The following theorem establishes conditions for the closure of the LSA-property.

Theorem 2.2:

Let $\Delta_i:\mathcal{X}\to T\mathcal{X}$ be C^∞(ID) that is LSA(Ω_i), where $(x_0,u_0)\in\Omega_i$ and $\rho_i := \{x\in\mathcal{X}\,|\,\exists u\in\mathcal{U},\ (x,u)\in\Omega_i\}$, $(i = 1,2)$.

If there exist nbhds. $W\subset\rho_1\cap\rho_2$ of x_0 and $\Omega'\subset\Omega_1\cap\Omega_2$ of (x_0,u_0) such that

(a) For $S := V[\Delta_1]\vee V[\Delta_2]$ dim $\Delta_s(x) = c_1$ (constant), all $x\in W$, and (2.2a)

(b) If $\{U_\alpha\}$ is a countable cover of open sets for W with local C^∞ submersions $f_\alpha : U_\alpha \to \mathbb{R}^{n-c_1}$ defined such that $(df_\alpha)S(x) = 0$, $x\in U_\alpha$, then

(i) The matrix $[d_2(df_\alpha F(x,u)]$ has a generic rank C_2 on $\Omega'\cap(U_\alpha\times\mathcal{U})$ and (2.2b)

(ii) For all local C^∞ submersion $\bar{f}_\alpha:\tilde{U}_\alpha \to \mathbb{R}^{c_2}$ such that $(d\bar{f}_\alpha)S = 0 \quad x\in\tilde{U}_\alpha$ where rank $d_2[d\bar{f}_\alpha F(x,u)] =$ rank $d_2[df_\alpha F(x,u)]$, $(x,u)\in\Omega'\cap[[U_\alpha\cap\tilde{U}_\alpha]\times\mathcal{U}]$, the matrix $R_\alpha := [d_2\bar{\Phi}]^T([d_2\bar{\Phi}][d_2\bar{\Phi}]^T)^{-1}[d_2\bar{\Phi}]$ exists and has a constant rank c_2 on $(x,u)\in\Omega'\cap[[U_\alpha\cap\tilde{U}_\alpha]\times\mathcal{U}]$, where $\bar{\Phi} = d\bar{f}_\alpha F(x,u)$. (2.2c)

Then Δ_s is LSA($\Omega'\cap(W\times\mathcal{U})$) where $\Omega'\cap(W\times\mathcal{U})$ is a nbhd. of (x_0,u_0).

Observe that for linear systems the above conditions are automatically satisfied and this leads to the closure of the [A,B] invariance property.

A general constructive procedure to identify a unique local supremal C^∞(ID)$\Delta^*:U\to TU$ that is LSA(Ω^*) (if it exists) where $x_0\in U$, $(x_0,u_0)\in\Omega^*$ has been developed (in contrast to the abstract Alg. 1 given in [1]).

The Local Disturbance Decoupling Problem (LDDP)

Consider the system $\tilde{\Sigma} := (\mathbb{R}^n, \mathbb{R}^m, \mathbb{R}^r, \mathbb{R}^\sigma, F(x,u,q), \mathcal{S})$. Let (x_0, u_0, q_0) be the operating point. Let $\bar{Q}$ be a nbhd. of q_0 in $Q[= \mathbb{R}^\sigma]$. The vector field $F(x,u,q_0)$ is called a *nominal vector field for $\tilde{\Sigma}$* (NVF) and the family $E_{\bar{Q}} := \{F(x,u,q) \mid q \in \bar{Q}\}$ is called the *disturbance fields* for $\tilde{\Sigma}$.

Definition 2.6:

LDDP is *locally sufficient solvable* (LSS) if there exists Δ that is LSA(Ω) where Ω is a nbhd. of (x_0,u_0) and $J^0(E_{\bar{Q}})(x,u) \subseteq \Delta(x)$, $(x,u) \in \Omega$.

Theorem 2.3:

Let (x_0,u_0) be the operating point of $\tilde{\Sigma}$. If Δ^* the supremal LSA(Ω^*) exists at $q = q_0$, then LDDP is LSS if and only if $J^0(E_{\bar{Q}})(x,u) \subseteq \Delta^*(x)$, $(x,u) \in \Omega$.

Note that if a supremal LSA distribution does not exist, then there may be some ΔLSA(Ω) such that $J^0(E_{\bar{Q}})(x,u) \subset \Delta(x)$

2.2 Local Weak Controllability

The distribution $\bar{\Delta}_c : \Omega \to T\mathcal{U}$ represents the extra degrees in control freedom left after implementing some local dynamic feedback map $\Gamma \in \{\Gamma_\alpha\}$ such that $\bar{\Delta}^1$ is $G_{(F,\Gamma)}$-invariant, $(x,u,) \in W_\alpha$, which may be used to further modify the system dynamics without destroying this local invariance.

Definition 2.7:

Let Δ be LSA(Ω), Ω is a nbhd. of (x_0,u_0) in $\mathcal{X} \times \mathcal{U}$ and $X_\alpha := \{X := (F(x,u), \Gamma^\alpha(x,u)) \in V(W_\alpha) \mid \Gamma^\alpha := \Gamma_0^\alpha + \Gamma_i^\alpha, \Gamma_i^\alpha \in \{\overline{\Gamma^\alpha}\}\}$.

Define the sequence of vector field families on $W_\alpha \subset \mathcal{X} \times \mathcal{U}$.

$$\mathcal{V}_\alpha^0 := \mathcal{S}_\alpha^0 := V[\bar{\Delta}_c]$$

$$\mathcal{V}_\alpha^{\mu+1} := \bigcup_{X \in X_\alpha} [X, \mathcal{S}_\alpha^\mu], \quad \mathcal{S}_\alpha^{\mu+1} := [\mathcal{S}_\alpha^\mu, \mathcal{V}_\alpha^{\mu+1}]^*, \quad \mu = 0,1,2,.,...$$

For some $\mu = k$, $\Delta^{cL}(x,u) = \Delta_{\mathcal{S}_\alpha^k}(x,u) = \Delta_{\mathcal{S}_\alpha^{k+1}}(x,u)$, all $(x,u) \in W_\alpha$. It is easily shown that $[\mathcal{S}_\alpha^k, V[T\mathcal{U}]]^* = \mathcal{S}_\alpha^k \cup V[T\mathcal{U}]$, all $(x,u) \in W_\alpha$. If $\dim \Delta_{\mathcal{S}_\alpha^k}(x,u) = c_1 =$ constant, and $\dim \Delta_{\mathcal{S}_\alpha^k \cup V[T\mathcal{U}]}(x,u) = c_2 =$ constant on $V_\alpha \subset W_\alpha$, then $\Delta_{\mathcal{S}_\alpha^k} : V_\alpha \to TV_\alpha$ is called a *local generalized dynamic decomposition on V_α* (LGDD(V_α)). Let $\Omega' \subset \Omega$ be the largest nbhd. of (x_0,u_0) such that $\dim \Delta_{\mathcal{S}_\alpha^k}(x,u) = c_1$, $(x,u) \in V_\alpha$ and $\cup\{V_\alpha\} = \Omega'$, this implies that there exists a C^∞-ID $\Delta^{cL} : \Omega' \to T\Omega'$ called a LGDD(Ω').

Proposition 2.1:

In Definition 2.7, $\Delta^{cL}(x,u) \subseteq \bar{\Delta}'(x,u)$, $(x,u) \in \Omega'$ and $V[\Delta^{cL}] \wedge V[T\mathcal{U}] = V[\bar{\Delta}_c]$.

Theorem 2.4:

$\Delta^{cL}:\Omega' \to T\Omega'$ is the first associated distribution of a C^∞-ID $\Delta'':\rho' \to T\rho'$, $\rho' := \{x \in X \mid u \in \mathcal{U},\ (x,u) \in \Omega'\}$ that is LSA(Ω'), and $\dim \Delta''(x) = c_1 - (m - \ell)$.

Proposition 2.2:

Let $\{U_\alpha\}$ be a countable cover of open sets for Ω' where Δ^{cL} is LGDD(Ω'). Let X_α be the family of all local vector fields defined on U_α such that for every $x \in X_\alpha$, $[X, V_{\bar{\Delta}'}](x,u) \subset V_{\bar{\Delta}'}(x,u)$ and $[X, V_{\Delta^{cL}}](x,u) \subset V_{\Delta^{cL}}(x,u)$, $(x,u) \in U_\alpha$. For all flows of X_α restricted to ', the reachable set at time T, $L_{(x_0,u_0)}(X_\alpha, T)$ has a nonempty interior on $G^T(X_\alpha, T) \cap \Omega' \subset \Omega'$.

Definition 2.8:

By Proposition 2.2, it is clear that Δ^{cL} is locally weakly controllable on Ω', hence, " a distribution Δ, LSA(Ω) is called locally weakly controllable on Ω, LWC(Ω), if $\bar{\Delta}^1:\Omega \to \Omega$ is LGDD(Ω).

Theorem 2.5:

Let Δ_i be LWC(Ω_i), ($i \in \underline{2}$). If $\tilde{V} := V[\Delta_1] \vee V[\Delta_2]$ & $\Delta_{\tilde{V}}$ is LSA(Ω), $\Omega \subset \Omega_1 \cap \Omega_2$, then $\Delta_{\tilde{V}}$ is LWC(Ω).

A general constructive procedure to identify supremal LWC distribution Δ^{Lc*} contained in some Δ LSA(Ω) distribution has been developed which does not require the knowledge of local feedback maps as in Definition 2.7.

2.3 Compatibility

Definition 2.9:

Let Δ_i be LSA(Ω_i) with 1st associated distribution $\bar{\Delta}_i^1:\Omega_i \to T\Omega_i$ and control distribution $\bar{\Delta}_{ci}:\Omega_i \to T\mathcal{U}$, where $(x_0,u_0) \in \Omega_i$, $(i \in \underline{k})$. The family $\{\Delta_1,\ldots,\Delta_k\}$ is said to be locally compatible on $\Omega \subset \bigcap_{i=1}^{k} \Omega_i$ a nbhd. of (x_0,u_0) if

(1) There exists local dynamic feedback maps $\Gamma_\beta:W_\beta \to T\mathcal{U}$, $W_\beta \in \Omega$, $\cup W_\beta = \Omega$ such that $[(F,\Gamma_\beta), V_{\bar{\Delta}_i^1}(x,u)] \subset V_{\bar{\Delta}_i^1}(x,u)$, $(x,u) \in W_\beta$ all W_β.

(2) If $V^0 := \bigcap_{i=1}^{k} V[\Delta_i]$, is LSA($\Omega$).

(3) $\bigvee_j V[\Delta_j] = \bigcup_j V[\Delta_j]$ and $\bigvee_j V[\bar{\Delta}_j^1] = \bigcup_j V[\bar{\Delta}_j^1]$ (By Theorem 1.5)

(4) The dimensions of all the above distributions are constant on Ω.

Theorem 2.6:

The family $\{\Delta_1,\ldots,\Delta_k\}$ in Definition 2.9 is locally compatible on Ω if and only if

(a) $\bigvee_j V[\Delta_j] = \bigcup_j V[\Delta_j]$, and $\bigvee_j V[\bar{\Delta}_j^1] = \bigcup_j V[\bar{\Delta}_j^1]$ $(x,u) \in \Omega$ (*)

where $\dim \bigcup_j V[\Delta_j] = \text{constant}$, $\dim \bigcup_j V[\bar{\Delta}_j^1] = \text{constant}$ $(x,u)\in\Omega$.

(b) $\dim \{\bigcup_j V[\bar{\Delta}_j^1]\}(x,u) = \dim \{\bigcup_j V[\Delta_j]\} + \dim \{\bigcup_j V[\bar{\Delta}_{cj}]\}$ $(**)$

for all possible unions.

Theorem 2.6 is fundamental when dealing with multiple decompositions and a weaker version applies for linear systems where 1,2,3 and 4 are automatically satisfied.

The Local Decoupling Problem (LDP)

Consider the system $\hat{\Sigma} := (\mathbb{R}^n, \mathbb{R}^m, (\mathbb{R}^{r_1} \times \mathbb{R}^{r_2} \times \ldots \times \mathbb{R}^{r_k}), F(x,u), (\delta_1, \delta_2, \ldots, \delta_k))$. The output maps $\delta_j : \mathbb{R}^n \to \mathbb{R}^{r_j}$ $(j \in \underline{k})$ are C^∞ submersions and all maps $(\delta_{\alpha_1}, \ldots, \delta_{\alpha_\ell}) : \mathbb{R}^n \to \mathbb{R}^{\alpha_1} \times \mathbb{R}^{\alpha_2} \ldots \times \mathbb{R}^{\alpha_\ell}$ $\alpha_\ell \in \{r_1, \ldots, r_1\}$ and $(\ell \in \underline{k})$.

Definition 2.10:

LDP is LSS if there exist $\{\Delta_1, \ldots, \Delta_k\}$ that are $\mathrm{LWC}(\Omega_1), \ldots, \mathrm{LWC}(\Omega_k)$ and are locally compatible on (Ω) $\Omega \subset \bigcap_{i=1}^{k} \Omega_k$, a nbhd. of (x_0, u_0) such that

(i) $\Delta_i(x) \subseteq \bigcap_{j \neq i} \operatorname{Ker} d\delta_j(x)$ $(x,u)\in\Omega$

(ii) $\Delta_i(x) + \operatorname{Ker} d\delta_i(x) = TX_x$ $(x,u)\in\Omega$. $(i \in \underline{k})$.

Assumption 2.1:

Assume that all Δ_i^{*cL} the supremal $\mathrm{LWC}(\Omega_i^*)$ such that $\Delta_i^{*cL}(x) \subseteq \bigcap_{j \neq i} \operatorname{Ker} d\delta_j(x)$ exist, and are locally compatible on $\Omega \subset \bigcap_{j=1}^{k} \Omega_j^*$.

Theorem 2.7:

Subject to Assumption 2.1, LDP is LSS if and only if

$$\Delta_i^{*cL}(x) + \operatorname{Ker} d\delta_i(x) = TX_x \quad (x,u)\in\Omega .$$

Clearly there is no reason that Δ_i^{*cL} should exist for all $(i \in \underline{k})$, and even if they do, they need not satisfy equations (*) and (**). There is at present no way of determining whether there exist a family of locally compatible distributions Δ_i that are LWC and satisfy both (*) and (**) in Theorem 2.6.

By analogy with the linear case [4] it would seem that a local dynamic extension might solve the decoupling problem provided (*) holds. A preliminary investigation shows that in general a local dynamic extension fails for nonlinear systems, and a closer analysis shows that linearity plays a crucial role in the success of extensions for linear systems. Further research is required to determine conditions under which local extensions may solve the local decoupling problem.

The local stabilization of the system about an operating point while conserving the local decompositions has been investigated under certain conditions. The results however are not reported here.

Acknowledgements

I would like to thank Professor W.M. Wonham for supervising this research.

Notation:

In this paper X, U, Z are open sets in $\mathbb{R}^n$, $\mathbb{R}^m$ and $\mathbb{R}^r$. This can be viewed as a local representation of the dynamic system Σ, where $F(x,u)$ and δ are the local forms of the system vector field and output map respectively.

References

1. S.H. Mikhail, W.M. Wonham, "Local Decomposability and the Disturbance Decoupling Problem in Nonlinear Autonomous Systems", Proc. 16th Annual Allerton Conference on Communications, Control and Computing, 1978, pp. 664-669.
2. W.M. Boothby, An Introduction to Differentiable Manifolds and Riemannian Geom., Academic Press, 1975.
3. Brickell and Clark, Differential Manifolds, Van Nostrand, 1970.
4. W.M. Wonham, Linear Multivariable Control: A Geometric Approach, Springer-Verlag, N.Y., second edition, 1979.
5. R. Hermann, A.J. Krenner, "Nonlinear Controllability and Observability", IEEE Trans. on Automatic Control, Vol. AC-22, No. 5, October 1977.
6. S.H. Mikhail, Ph.D. Thesis (in preparation), Dept. of Electrical Engineering, University of Toronto.

TOPOLOGICAL CONSIDERATIONS IN SYSTEMS REALIZATION*

REN JENG SU, T. J. TARN and DAVID L. ELLIOTT
Department of Systems Science and Mathematics
Washington University
St. Louis, Missouri 63130

ABSTRACT

In the set-theoretic and linear cases, the relations between input - output and state representations are now understood. Here we attempt, without assuming linearity, to study the relationships of the topologies of the input, output, and state spaces.

I. INTRODUCTION

This investigation has been motivated by Kalman's realization theory for linear systems [1], and those attempts to provide a unified framework for the study of system theory such as Zadeh and Desoer [2], Mesarovic and Takahara [3], d'Alessandro [4], Balakrishinan [5] and some others. We are concerned with the topological aspects of the general theory of state space model building along the direction of realization theory, i.e. from input-output model to state space model.

Viewing a "black box" with inputs and outputs as "a system", the behavioral data of the black box involving inputs and outputs are usually called the external description of the system. An internal description is a postulated internal mechanism which can, to some extent, represent the external description. Given a system by way of specifying its external description, the problem of postulating and inducing an internal description is called realization problem (see Kalman [1]).

It is conceived that with more analytic informations about the external behavior, one can have more sophisticated internal mechanism postulated. When the sets of inputs and outputs of a system are viewed as purely abstract sets without any other mathematical structures, a system can be defined as follows.

Definition 1. Given an input set U and an output set Y, a system S is defined as a relation of U and Y, i.e. S is a system if $S \subset U \times Y$. (Conventionally, domain of S being equal to U is always assumed.)

With this primitive concept of external descriptions, a notion termed states of a system can be defined as parametrization of the general relation into functional type relation.

Definition 2. Let $S \subset U \times Y$ be a system. A state of S is a function $f : U \to Y$ with the property that $f \subset S$. A state representation of S is a family of states with Σ as index set, $\{f_\sigma : U \to Y\}_{\sigma \varepsilon \Sigma}$, such that $\bigcup_{\sigma \varepsilon \Sigma} f_\sigma = S$.

The existence problem of this state representation has been studied and a regrous proof of the following theorem can be found in d'Alessandro [4].

Theorem 1. For any system S there exists a state representation.

In this paper, we shall consider two different ways of sophisticating the system behavior by adding some mathematical structures on the input and output sets and try to postulate and to induce two kinds of state representations which are based on the notion of definition 2 but with richer structures. In section II, the input and output sets are topological spaces and in section III the concept of time is taken into account.

---

*This work was supported partially by the National Science Foundation Grant Nos. ENG 76-80365A01, ENG 78-22166 and INT-7902976.

II. TOPOLOGICAL SYSTEMS

In this section the input and output sets, U and Y respectively, are both topological spaces and a system is a subset of $U \times Y$ as in definition 1.

For a system of this type, the following notion of state representations comes naturally.

Definition 3. A continuous state of a system $S \subset U \times Y$ is a continuous function $f:U \rightarrow Y$ which is contained in S. A continuous state representation of S is a family of continuous states $\{f_\sigma : U \rightarrow Y\}_{\sigma \varepsilon \Sigma}$, where Σ is an index set, such that $\bigcup_{\sigma \varepsilon \Sigma} f_\sigma = S$.

It is clear that not every (topological) system has a continuous state representation. What conditions on a system can guarantee that there exists one is the major concern of this section.

At first, let us observe a simple fact: A system $S \subset U \times Y$ has a continuous state representation if and only if for each $(u,y) \varepsilon S$, there exists a continuous function $f:U \rightarrow Y$ such that $f(u)=y$ and $f \subset S$.

Before stating a necessary condition we need the following notion of continuity.

Definition 4. For each $u \varepsilon U$, let $S(u)$ denote the set $\{y \varepsilon Y |\ (u,y) \varepsilon S\}$. A system is said to be lower semi-continuous if for every open set V in Y, the set $\{u \varepsilon U | s(u) \cap V \neq \phi\}$ is open in U.

With this notion, we have a necessary condition.

Theorem 2. If a system $S \subset U \times Y$ admits a continuous state representation then S is lower semi-continuous.

Proof. Suppose S has a continuous state representation. Firstly, we will show $\{u \varepsilon U | S(u) \cap V \neq \phi\} = \bigcup\{f^{-1}(V) \mid f \text{ is a continous function } :U \rightarrow Y, \text{ and } f \subset S\}$. By the fact stated above, the first set includes the second and the other direction of inclusion is obvious. Thus, the equality holds. Now, because of the continuity of each f, every $f^{-1}(V)$ is open in U and so is the union. This completes the proof. □

In general, lower semi-continuity is not a sufficient condition for the existence of a continuous state representation. The following is a counter example.

Example. Let $U = [0,1]$, and $Y = \mathbb{R}$, the set of reals. A system $S \subset U \times Y$ is defined as:

$$S: u \mapsto \begin{cases} \{0\} & u \varepsilon [0,1/3] \\ \{0,1\} & u \varepsilon (1/3,2/3) \\ \{1\} & u \varepsilon [2/3,1] \end{cases}$$

It is easy to check that S is lower semi-continuous. But there is obviously no continuous function from [0,1] into $\mathbb{R}$ which is in S. □

Concerning the notion of lower semi-continuity, the following observation is useful for our purpose.

Lemma 1. Let $S \subset U \times Y$ be a lower semi-continuous system with the topology of U being T_1. For any $(u_0,y_0) \varepsilon S$, let $\tilde{S}(u_0,y_0) \subset S$ be defined as:

$$\tilde{S}(u_0,y_0): (u) \mapsto \begin{cases} \{y_0\}, & u = u_0 \\ S(u), & u \neq u_0, \end{cases}$$

then $\tilde{S}(u_0,y_0)$ is also lower semi-continuous.

Proof. For any open set V in Y, the set $\{u|\tilde{S}(u_0,y_0)(u)\cap V\neq\phi\}$ is either equal to $\{u|S(u)\cap V\neq\phi\}$ or $\{u|S(u)\cap V\neq\phi\}\setminus\{u_0\}$. Since U is a T_1-space, both sets are open in U by the l.s.c. of S. Thus $\tilde{S}(u_0,y_0)$ is also l.s.c. □

We notice that within some special category of systems the existence of continuous state representation is directly implied by the continuous selection theorem due to E. Michael which is stated as follows without proof. The interested readers are referred to Michael [6].

Theorem 3. Let U be a paracompact space, and Y be a Banach space. If S is a lower semi-continuous system with S(u) being convex and closed in Y for every $u\varepsilon U$, then there exists one continuous state f: $U\to Y$ which is in S. Finally we have:

Theorem 4. For every system satisfying the conditions of theorem 3, there exists a continuous state representation.

Proof. From lemma 1, $\tilde{S}(u_0,y_0)$ induced from the system S is l.s.c. Moreover, $\forall u\varepsilon U$, $\tilde{S}(u_0,y_0)(u)$ is convex and closed in Y. Then, from Theorem 3, there exists a continuous function f: $U\to Y$ which is a continuous state of $\tilde{S}(u_0,y_0)$. It is of course a continuous state of S. What we have shown is that for every $(u_0,y_0)\varepsilon S$, there exists a continuous state of S. By the simple fact stated previously, the system admits a continuous state representation.

□

III. TIME SYSTEMS

The term "time systems" is generally applied to those systems whose inputs and outputs are time functions. In this paper a time function means a function from a subset of $[0,\infty)$ into some abstract set. For an arbitrary set A, A_t will denote the set of all functions from $[t,\infty)$ into A, where $t\geq 0$, and A^t will denote the set of all functions from $[0,t)$ into A, where $t > 0$.

Definition 5. Given two sets A and B, a time system is a system (in the sense of definition 1) $S_t\subset U_t\times Y_t$ with $U_t\subset A_t$ and $Y_t\subset B_t$, where $t\varepsilon[0,\infty)$ is called the initial time of S_t.

Let us observe that a time system is also itself a family of time functions from $[t,\infty)$ into A × B. Moreover, from a time system S_t with initial time t a family of time systems indexed by initial times $t'\geq t$ can be derived by defining

$$S_{t'}\triangleq\{(u|_{[t',\infty)},y|_{[t',\infty)})|(u,y)\varepsilon S_t\}$$

These time systems will be called subsystems derived from S_t. Since a time system is a system (definition 1), it admits a state representation (definition 2) which will be called an initial state representation in this case. At this point it is natural to ask that how to construct a family of initial state representations for a given system and its derivative subsystems respectively such that there are certain connections among the initial states at different times.

In the rest of this section a simple case is constructed for demonstration. The following is four types of operations needed on time functions.

(1) Cutting.

$\forall$ t, $t'\varepsilon[0,\infty)$, $t<t'$, a map $T_{t'}$: $A_t\to A_{t'}$ defined by

$T_{t'}: f \mapsto f|_{[t',\infty)}$ is called a cutting operator at t' on A_t.

(2) Truncation.

A map $T^{t'}: A^{t} \to A^{t'}$ defined by

$\forall\, t, t' \in [0,\infty),\ t' < t$

$T^{t'}: f \mapsto f|_{[0,t')}$ is called a truncating operator at t' on A^{t}.

(3) Shifting.

$\forall\, t, t' \in [0,\infty)$, a map $F^{t\to t'}: A_t \to A_{t'}$ defined by

$F^{t\to t'}(f)(\tau) = f(\tau - t + t')$ is called a shifting operator from t to t' on A_t.

(4) Concatenation.

$\forall\, t \in [0,\infty)$, a map $*_t: A^{t} \times A_t \to A_0$ defined by

$*_t: (f,g) \mapsto h$ where $h(\tau) = \begin{cases} f(\tau), & 0 \le \tau < t \\ g(\tau), & t \le \tau < \infty \end{cases}$

is called a concatenation operator at t.

With these operations the notions of time-invariance and closedness under concatenations are defined.

Let A_t be a family of time functions with initial time t.

Definition 6. A_t is said to be time-invariant if

$\forall\, \tau, \tau' \in [t,\infty),\ F^{\tau\to\tau'}(T_\tau(A_t)) \subset T_{\tau'}(A_t)$.

Definition 7. A_t is said to be closed under concatenations if

$\forall\, \tau > t,\ *_\tau(T^{\tau}(A_t) \times T_\tau(A_t)) \subset A_t$.

Definition 8. A time system $S_t \subset U_t \times Y_t$ with initial time t is said to be time invariant (closed under concatenation) if S_t is time-invariant (closed under concatenation) as a family of time functions.

Note: For a time-invariant family of time functions A_t, the map $F^{\tau\to\tau'}: T_\tau(A_t) \to T_{\tau'}(A_t)$ is in fact a one - to - one and onto map. Thus we can identify all the families $T_\tau(A_t)$, $\tau \geq t$. With this understanding we will use a single notation $\bar{A}$ for every $T_\tau(A_t)$ hereon whenever A_t is time-invariant.

For the time systems with time-invariance and/or closedness under concatenations we postulate two types of state representations.

Definition 9. Let $S \subset U \times Y$ be a time system with U and Y being time-invariant. A family of input-output functions $\{f_\sigma: U \to Y\}_{\sigma\in\Sigma}$ is called over-all state representation for S if

$\bigcup_{\sigma\in\Sigma} f_\sigma = S_t$, where S_t is any subsystem derived from S.

Definition 10. Let $S \subset U \times Y$ be a time system with the input set U and Y being time-invariant and closed under concatenations. A family of input-output functions $\{f_\sigma: U \to Y\}$ is called a state space representation for S if

(i) $\bigcup_{\sigma\varepsilon\Sigma} f_\sigma = S_t$, for every subsystem S_t derived from S and

(ii) there exists a map ϕ: $\Sigma \times U \times [o,\infty) \rightarrow \Sigma$ with the properties that:

(a) for any $\sigma\varepsilon\Sigma$, $\phi(\sigma,u,0) = \sigma$, $\forall u \varepsilon U$, and

(b) for any $\sigma\varepsilon\Sigma$, $u\varepsilon U$ and $t\varepsilon[0,\infty)$ let

$\phi(\sigma,u,t) = \sigma_t$ then

$f_{\sigma_t}(\omega) = T_t(f_\sigma(u *_t \omega))$ $\forall \omega\varepsilon U$

Here are the main theorems of this section.

Theorem 5. A time-invariant system admits an over-all state representation.

Proof. Any initial state representation for a time-invariant system is an over-all state representation. □

Theorem 6. A system which is time-invariant and closed under concatenation admits a state space representation.

Proof. Let $\{f_{\sigma'}: U \rightarrow Y\}_{\sigma'\varepsilon\Sigma'}$ be an initial state representation for S. Define

$\tilde{\Sigma} := \{(\sigma',u,t) \mid \sigma'\varepsilon\Sigma', u\varepsilon U, t\varepsilon[0,\infty)\}$ and define

F: $\tilde{\Sigma} \rightarrow Y^U$ by

$$(F(\sigma',u,t))(\omega) = \begin{cases} T_t(f_{\sigma'}(u *_t \omega)), & t > 0 \\ f_{\sigma'}(\omega), & t = 0. \end{cases}$$

Thus, every $(\sigma',u,t)\varepsilon\Sigma'$ corresponds to an initial state of S_t. Since we have identified S_t and S, (σ',u,t) corresponds to an initial state of S. Now define an equivalence relation in $\tilde{\Sigma}$ by

$$(\sigma'_1,u_1,t_1) \sim (\sigma'_2,u_2,t_2) \text{ if } F(\sigma'_1,u_1,t)=F(\sigma'_2,u_2,t_2).$$

And let $\Sigma \triangleq \tilde{\Sigma}/\sim$, then $\forall\sigma\varepsilon\Sigma$, it corresponds to a unique map f_σ: $U \rightarrow Y$ and $f_\sigma \subset S$, namely an initial state of S. It is also clear that $\bigcup_{\sigma\varepsilon\Sigma} f_\sigma = S$. Therefore $\{f_\sigma\}_{\sigma\varepsilon\Sigma}$ is an overall state representation. Now define a map ϕ: $\Sigma \times U \times [0,\infty) \rightarrow \Sigma$. For each $\sigma\varepsilon\Sigma$, it is an equivalence class in $\tilde{\Sigma}$. Let (σ'_1,u_1,t_1) be an arbitrary element in the class σ.

Define $\phi(\sigma,\omega,\tau) = [(\sigma'_1,u_1 *_{t_1} u_2,t_1 + t_2)]$, where the brackets denote an equivalence class in $\tilde{\Sigma}$, namely an element in Σ. It can be checked that if $(\sigma'_1,u_1,t_1) \sim (\sigma'_2,u_2,t_2)$ in $\tilde{\Sigma}$ then

$$(\sigma'_1,u_1 *_{t_1} \omega_1,t_1 + \tau) \sim (\sigma'_2,u_2 *_{t_2} \omega,t_2 + \tau).$$

Therefore ϕ is a well-defined map from $\Sigma \times U \times [0,\infty)$ into Σ. Finally, the properties (i) and (ii) follow from the definition and the proof is completed. □

IV. EPILOGUE

The underlying idea of realization problem in this paper is from Kalman (see Kalman [1]). We have adopted the concept of "states" suggested by Zadeh and Desoer [2] and are trying to understand how such "state space" are impacted by the structures of spaces of inputs and outputs. Our special interests are in the cases when both topologies and time concept come to play roles. An inspiring paper by d'Alessandro [4] is very close to our point of view. In that paper d'Alessandro stated: let Σ be a state representation (not necessarily continuous) for a system S, then it is a continuous state representation if and only if the topology of the input space is stronger than the Σ-projective topology.

Because of our realization-problem-oriented point of view, we asked the existence problem with topologies of inputs and outputs given in advance. This leads to the nontriviality of the problem. Balakrishnan [5] and Sussman [7] are examples among the works with indirect touch upon the relationships between input topologies and state spaces. For the time system, a detailed study on set-theoretic level can be found in the book by Mesarovic and Takahara [3].

For nonlinear systems on general topological state spaces, the notions of causality, state transition map, output generating map, etc., require a more careful treatment than those given in the literature for the linear and set-theoretic cases. These issues will be addressed in another paper.

V. REFERENCES

(1) R. E. Kalman, "Realization theory of linear dynamical systems", in Control Theory and Topics in Functional Analysis, International Atomic Energy Agency, Vienna, 1976.

(2) L. A. Zadeh and C. A. Desoer, "Linear system theory", McGraw Hill, New York, 1963.

(3) M. D. Mesarovic and Y. Takahara, "General system theory: Mathematical foundations", Academic Press, New York, 1975.

(4) P. d'Alessandro, "The Concept of state and axiom of choice", J. of the Franklin Institute, Vol. 304, No. 2/3, 1977.

(5) A. V. Balakrishnan, "On the state-space theory of nonlinear systems", Functional analysis and optimization, Academic Press, New York, 1966.

(6) E. Michael, "Continuous selections I", Ann. of Math., Vol. 63, 1956.

(7) H. J. Sussmann, "Semigroup representations, bilinear approximation of input-output maps, and generalized inputs", Mathematical Systems Theory, Lecture Notes in Economics and Mathematical Systems, Vol. 131, Springer-Verlag, New York, 1976.

ASYMPTOTIC STABILITY IN PARTIALLY DISSIPATIVE NONLINEAR SYSTEMS

R. K. MILLER
Department of Mathematics
Iowa State University
Ames, Iowa 50011

A. N. MICHEL
Department of Electrical Engineering
and Engineering Research Institute
Iowa State University
Ames, Iowa 50011

ABSTRACT

In a recent paper by the authors, necessary and sufficient conditions were established in order that the addition of damping (at random locations) to a linear conservative dynamical system ensures the asymptotic stability of the equilibrium. In the present paper, we extend these results to nonlinear systems. For the linear case, necessary and sufficient conditions for asymptotic stability were established, while for the nonlinear case, sufficient conditions and also some necessary and sufficient conditions for asymptotic stability are obtained.

I. INTRODUCTION

In a recent paper [7], necessary and sufficient conditions were established in order that the addition of damping (at random locations) to a linear conservative dynamical system ensures the asymptotic stability of the equilibrium. In the present paper, we extend these results to nonlinear systems. For the linear case, necessary and sufficient conditions for asymptotic stability were established (in [7]), while for the nonlinear case, sufficient conditions and also some necessary and sufficient conditions for asymptotic stability are obtained (in Section 3 of the present paper).

It is emphasized that the application of the present results to specific problems will usually not require a search for appropriate Lyapunov functions. Indeed, a stability analysis by the present method involves the following two steps:

(a) given a system with dissipation, the stability of its trivial solution (equilibrium) is ascertained by determining the stability of the associated conservative system, i.e., by determining whether the potential energy is a minimum at the equilibrium; and

(b) attractivity of the equilibrium of the entire system (with dissipation) is determined from the system topology (system configuration).

In addition to [7], results related to the present ones have been reported in [8] and [9]. Specifically, the results in [8], which are only applicable to linear systems, are a special case of a result reported herein. The work in [9], which deals with a class of nonlinear circuits, does not overlap with the present results but has a similar flavor.

The present approach to stability analysis appears to be new. Furthermore, since the present method involves concepts from control theory (namely, the notion of observability and system distinguishability), these results provide further insight into the mechanisms of stability and stabilization.

II. LINEAR HAMILTONIAN SYSTEMS

The Hamiltonian formulation in mechanics is well established (see, e.g., [2,10]). Given a Hamiltonian function $H(q_1,\ldots, q_n, p_1,\ldots, p_n)$, the Hamiltonian differential equations for conservative systems are

$$\left.\begin{aligned} \dot{q}_i &= \frac{\partial H}{\partial p_i}(q_1,\ldots, q_n, p_1,\ldots, p_n) \\ \dot{p}_i &= \frac{-\partial H}{\partial q_i}(q_1,\ldots, q_n, p_1,\ldots, p_n), \end{aligned}\right\} \tag{1}$$

$i=1,\ldots, n$, or in vector notation,

$$\begin{aligned} \dot{q} &= \frac{\partial H}{\partial p}(q,p) \\ \dot{p} &= \frac{-\partial H}{\partial q}(q,p), \end{aligned}$$

where the q_i denote generalized position coordinates, the p_i denote the generalized momentum coordinates and H represents the total energy of a system. (Using appropriate analogies, lossless electrical systems, electromechanical systems, etc., can be represented by (1) as well.) The motions of system (1) are always such that the energy is conserved, since

$$\frac{d}{dt}H(q,p) = \sum_{i=1}^{n}\left(\frac{\partial H}{\partial q_i}\dot{q}_i + \frac{\partial H}{\partial p_i}\dot{p}_i\right) = \sum_{i=1}^{n}\left(\frac{\partial H}{\partial q_i}\frac{\partial H}{\partial p_i} - \frac{\partial H}{\partial p_i}\frac{\partial H}{\partial q_i}\right) = 0.$$

A linear Hamiltonian differential equation is obtained from a quadratic Hamiltonian of the form

$$H(q,p) = \frac{1}{2}q^TH_1q + \frac{1}{2}p^TH_2p \tag{2}$$

where we can assume, without loss of generality, that the matrices H_1 and and H_2 are symmetric. Since in general $\frac{dH}{dt} \equiv 0$, we can use H as a Lyapunov function for (1) (when the potential energy has a local isolated minimum at the equilibrium $(q^T,p^T) = (0^T,0^T)$.) In particular, in the linear case (2) we ensure stability of the trivial solution with the assumptions that H_1 and H_2 are positive definite (see, e.g., [3]).

The linear system of differential equations corresponding to the Hamiltonian (2) is given by

$$\left.\begin{aligned} \dot{q} &= H_1p \\ \dot{p} &= -H_2q. \end{aligned}\right\} \tag{3}$$

Now if viscous damping is added, then system (3) will be replaced by

$$\left.\begin{aligned} \dot{q} &= H_1p \\ \dot{p} &= -H_2q + Kp, \end{aligned}\right\} \tag{4}$$

where $B_1 \triangleq H_2K + K^TH_2$ is negative semidefinite. In this case, the derivative of H(q,p) along the solutions of (4) is given by

$$D_{(4)}H = p^TB_1p \leq 0.$$

If in particular we specialize (2) to a simple mechanical system consisting of n rigid bodies with masses m_i, $i = 1,\dots, n$, then (2) will assume the form

$$H(q,p) = \frac{1}{2} q^T H q + \frac{1}{2} p^T M^{-1} p, \tag{5}$$

where $M = \text{diag}\,[m_1, m_2,\dots, m_n]$ and $H = H^T$ (which characterizes the potential energy term) is determined by the system configuration. For this case Eq. (4) assumes the form

$$\left.\begin{aligned} \dot{q} &= M^{-1} p \\ \dot{p} &= -Hq + KM^{-1} p \end{aligned}\right\} \tag{6}$$

which is equivalent to Newton's Law ($p = M\dot{q}$, $K = K^T$),

$$M\ddot{q} + Hq - K\dot{q} = 0.$$

The derivative of (5) along the solutions of (6) is easily computed to be

$$D_{(6)} H = p^T M^{-1} K M^{-1} p.$$

Our first result, which is similar to a result reported in [9], was proved in our earlier paper [7] and is restated here for purposes of completeness.

Theorem 1. Consider the Hamiltonian (5) and the system (6) with $M = \text{diag}[m_1, m_2,\dots,m_n] > 0$, $H = H^T > 0$, $K = K^T \leq 0$. Then the trivial solution of (6) is asymptotically stable if and only if the pair $(K, M^{-1}H)$ is observable[1]. ■

III. NONLINEAR HAMILTONIAN SYSTEMS

We now extend the result of Section II to nonlinear systems. To this end we consider a Hamiltonian of the form

$$H(q,p) = \frac{1}{2} p^T M^{-1} p + G(q) \tag{7}$$

and the associated system (with damping)

$$\left.\begin{aligned} \dot{q} &= M^{-1} p \\ \dot{p} &= KM^{-1} p - \nabla G(q) \end{aligned}\right\} \tag{8}$$

where $G: R^n \to R$ is assumed to be continuously differentiable over R^n and where ∇G denotes the gradient of G. We will find it convenient to associate with Eq. (8) an output equation of the form

$$y = \text{diag}[KM^{-1}, KM^{-1}] \begin{bmatrix} p \\ G(q) \end{bmatrix} . \tag{9}$$

Definition. System (8), (9) is called distinguishable (see, e.g., [5, p.377]) if whenever $(q(t), p(t))$ is a solution of (8) with $(q(0),p(0)) \neq (0,0)$, then the output $y(t) \not\equiv 0$. System (8), (9) is called locally distinguishable if there is an $\varepsilon > 0$ such that when $(q(t), p(t))$ is a solution of (8) and $0 < |p(0)| + |q(0)| < \varepsilon$, then the output $y(t) \not\equiv 0$.

We now prove the following result.

---

[1] Let U and V be matrices of dimensions m×n and n×n, respectively. We say that the pair (U,V) is observable if and only if the matrix $[U^T, (UV)^T, \dots, (UV^{n-1})^T]$ has full rank.

Theorem 2. Consider the Hamiltonian (7) and the system (8) with $M = \mathrm{diag}[m_1, m_2, \ldots, m_n] > 0$, G positive definite with respect to the origin, and $K = K^T \leq 0$. Then the trivial solution of system (8) is asymptotically stable if and only if the system (8), (9) is locally distinguishable.

Proof. Since G is positive definite, we may choose H given by (7) as a Lyapunov function. The derivative of H with respect to t along the solutions of (8) is given by

$$DH_{(8)}(q,p) = p^T M^{-1} K M^{-1} p \leq 0$$

for all (q,p). Let

$$N = \{(q,p) : KM^{-1}p = 0\}$$

be the null set of $DH_{(8)}$ and let N_I be the largest invariant subset of N. If $(q,(t), p(t))$ is a solution of (8) with $(q(0), p(0)) \varepsilon N_I$ and $|q(0)| + |p(0)|$ sufficiently small, then $KM^{-1}p(t) \equiv 0$ and

$$0 = KM^{-1}\dot{p}(t) = KM^{-1}(KM^{-1}p(t) - \nabla G(q(t)) = -KM^{-1}\nabla G(q(t)).$$

Since system (8), (9) is locally distinguishable, then $p(t) \equiv q(t) \equiv 0$. By the invariance theorem (see [4]) the trivial solution of (8) is asymptotically stable.

Conversely, assume that system (8), (9) is not locally distinguishable. Then in any neighborhood $\mathcal{U}$ of the origin (0,0) there is a nontrivial solution (q,p) of (8) which starts in $\mathcal{U}$ and for which the output (9) is identically zero. Thus (q,p) will also solve the stable, conservative Hamiltonian system given by

$$\dot{q} = M^{-1}p$$

$$\dot{p} = -\nabla G(q).$$

Since G is positive definite and since $M = M^T > 0$, then $H(q(t),p(t)) \equiv H(q(0),p(0)) \triangleq H_0 > 0$ and $(q(t),p(t)) \not\to 0$ as $t \to \infty$. ■

Essentially the same proof works for the next result.

Theorem 3. Consider the Hamiltonian (7) and the system (8) with $M = \mathrm{diag}[m_1, m_2, \ldots, m_n] > 0$, with G positive definite (with respect to the origin) for all q, with $G(q) \to \infty$ as $|q| \to \infty$ (i.e., G is radially unbounded), and with $K = K^T \leq 0$. Then the trivial solution of (8) is asymptotically stable in the large if and only if (8), (9) is distinguishable. ■

In certain cases the distinguishability of the nonlinear system (8), (9) is easily checked. For example, if G_1 is the linear part of ∇G at $q = 0$ so that

$$\nabla G(q) = G_1 q + o(q), \quad |q| \to 0,$$

then system (8), (9) can be linearized and we have

$$\left.\begin{aligned} \dot{q} &= M^{-1}p \\ \dot{p} &= KM^{-1}p - G_1 q \\ y^T &= [(KM^{-1}p)^T, (KM^{-1}G_1 q)^T]. \end{aligned}\right\} \qquad (10)$$

In this case we obtain the following result.

Corollary 1. Assume $M = \mathrm{diag}[m_1, m_2, \ldots, m_n] > 0$, $\nabla G(q) = G_1 q + o(q)$ near $q = 0$ with $G_1 = G_1^T > 0$ and $K = K^T \leq 0$. Then the trivial solution of (8) is asymptotically stable if $(K, M^{-1}G_1)$ is observable.

Proof. If $(K, M^{-1}G_1)$ is observable, then system (10) is observable (see the proof of Theorem 1 (given in [7]) and see, e.g., [5]). If system (10) is observable, then the corresponding nonlinear system (8), (9) must be locally observable (see [5, p.378]) and hence also distinguishable. Apply now Theorem 2 to complete the proof. ■

Remark. Theorems 1 to 3 and Corollary 1 can be stated in stronger terms by recalling the facts that (a) asymptotically stable plus autonomous imply uniformly asymptotically stable, and (b) asymptotically stable, autonomous and linear imply global exponential stability.

IV. AN EXAMPLE

Consider the linear mechanical mass-spring system of Fig. 1 which is governed by the equations

$$\left.\begin{aligned} m_1\ddot{x}_1 + k_1x_1 + k(x_1 - x_2) &= 0 \\ m_2\ddot{x}_2 + k_2x_2 + k(x_2 - x_1) &= 0, \end{aligned}\right\} \qquad (A)$$

where x_i denotes the displacement of mass m_i and k_1, k_2, k denote linear spring constants. When the initial state of this conservative system is displaced from its equilibrium position, the system will remain in motion indefinitely. If linear viscous damping is added at some or all of the masses and springs, as shown in Fig. 2, then the governing equations become

$$\left.\begin{aligned} m_1\ddot{x}_1 + k_1x_1 + k(x_1 - x_2) + B_1\dot{x}_1 + B(\dot{x}_1 - \dot{x}_2) &= 0 \\ m_2\ddot{x}_2 + k_2x_2 + k(x_2 - x_1) + B_2\dot{x}_2 + B(\dot{x}_2 - \dot{x}_1) &= 0 \end{aligned}\right\} \qquad (B)$$

where $B_1 \geq 0$, $B_2 \geq 0$, $B \geq 0$ and $B_1 + B_2 + B > 0$.

In applying Theorem 1 to this system, we let

$$K = \begin{bmatrix} (-B_1 - B) & B \\ B & (-B_2 - B) \end{bmatrix}, \qquad M^{-1}H = \begin{bmatrix} \frac{1}{m_1} & 0 \\ 0 & \frac{1}{m_2} \end{bmatrix} \begin{bmatrix} (k_1 + k) & -k \\ -k & (k_2 + k) \end{bmatrix},$$

and we consider the following possibilities:

Case 1: $\det K \neq 0$

Case 2a: $\det K = 0$ with $B_1 = B_2 = 0$

Case 2b: $\det K = 0$ with $B_1 = B = 0$

Case 2c: $\det K = 0$ with $B_2 = B = 0$.

For case 1, the pair $(K, M^{-1}H)$ is observable.

For case 2a, we have

$$K(M^{-1}H) = B\begin{bmatrix} \dfrac{-(k_1+k)}{m_1} - \dfrac{k}{m_2} & \dfrac{(k_2+k)}{m_2} + \dfrac{k}{m_1} \\ \dfrac{(k_1+k)}{m_1} + \dfrac{k}{m_2} & \dfrac{-(k_2+k)}{m_2} - \dfrac{k}{m_1} \end{bmatrix} .$$

In order that the pair $(K,M^{-1}H)$ not be observable, it is necessary and sufficient that the first row of $KM^{-1}H$ be the negative of the second row, i.e.,

$$\frac{k_1+k}{m_1} + \frac{k}{m_2} = \frac{k_2+k}{m_2} + \frac{k}{m_1} ,$$

or

$$\frac{k_1}{m_1} = \frac{k_2}{m_2} .$$

This is the condition under which it is possible for the two masses to move in synchronism.

For case 2b it is easy to compute that the condition which ensures that the pair $(K,M^{-1}H)$ be observable is $B_2k > 0$.

For case 2c the condition which ensures that the pair $(K,M^{-1}H)$ be observable is $B_1k > 0$.

Applying Theorem 1 we see that the system of Fig. 2 will be asymptotically stable for all the above cases, except case 2a.

Continuing this example, consider the system obtained from Fig. 2 by replacing the linear springs by nonlinear ones. Specifically, replace the linear spring restoring forces k_1u, k_2v, kw by $g_1(u), g_2(v)$ and $g(w)$, respectively, where $g_1 : R \to R, g_2 : R \to R$, and $g : R \to R$ are assumed to be differentiable. Then linear system (B) will be replaced by the nonlinear system

$$\left.\begin{aligned} m_1\ddot{x}_1 + g_1(x_1) + g(x_1 - x_2) + B_1\dot{x}_1 + B(\dot{x}_1 - \dot{x}_2) &= 0 \\ m_2\ddot{x}_2 + g_2(x_2) - g(x_1 - x_2) + B_2\dot{x}_2 - B(\dot{x}_1 - \dot{x}_2) &= 0 \end{aligned}\right\} \qquad (C)$$

where it is assumed that g_1, g_2 and g satisfy the conditions $g_1(0) = g_2(0) = g(0) = 0$ and $g_1'(x) > 0$, $g_2'(x) > 0$, and $g'(x) > 0$ for all $x \neq 0$. Once more it will be assumed that $B_1 \geq 0$, $B_2 \geq 0$, $B \geq 0$, and $B_1 + B_2 + B > 0$.

Next, to (C) we adjoin the outputs

$$\left.\begin{aligned} y_1 &= B\dot{x}_2 - (B + B_1)\dot{x}_1 \\ y_2 &= B\dot{x}_1 - (B + B_2)\dot{x}_2 \\ y_3 &= [(B+B_1)/m_1]\cdot[g_1(x_1) + g(x_1 - x_2)] \\ &\quad - (B/m_2)\cdot[g_2(x_2) - g(x_1 - x_2)] \\ y_4 &= -(B/m_1)\cdot[g_1(x_1) - g(x_1 - x_2)] \\ &\quad - [(B + B_2)/m_2]\cdot[g_2(x_2) - g(x_1 - x_2)]. \end{aligned}\right\} \qquad (D)$$

In studying the asymptotic stability of the trivial solution of system (C) we check when the system (C), (D) is distinguishable. We accomplish this by considering several cases.

Case 1. $B(B_1 + B_2) + B_1B_2 \neq 0$.

If all $y_i \equiv 0$, then by (D), $\dot{x}_1 \equiv \dot{x}_2 \equiv 0$. Also, $g_1(x_1) \equiv -g(x_1 - x_2) = -g_2(x_2)$. Since $x_1g_1(x_1) > 0$ if $x_1 \neq 0$ and $x_2g_2(x_2) > 0$ if $x_2 \neq 0$, then $x_1 \equiv x_2 \equiv 0$. Thus (C), (D) is distinguishable in this case and system (C) is asymptotically stable in the large.

Case 2. $B_1 > 0$, $B = B_2 = 0$.

If all $y_i \equiv 0$, then from (D) we see that $\dot{x}_1 \equiv 0$ and so $x_1 \equiv c_1$ is constant and $g_1(c_1) = -g(c_1 - x_2(t))$. Thus $x_2 = c_2 = -g^{-1}(-g_1(c_1)) + c_1$ is constant. The only constant solution of (C) is the trivial one. Thus system (C), (D) is distinguishable in this case and system (C) is asymptotically stable in the large.

Case 3. $B_2 > 0$, $B = B_1 = 0$.

Using an identical argument as in Case 2, it follows that system (C), (D) is distinguishable and system (C) is asymptotically stable in the large in this case.

Case 4. $B_1 = B_2 = 0$, $B > 0$.

This case is more complicated. If all $y_i \equiv 0$, then $\dot{x}_1 \equiv \dot{x}_2$ and

$$[g_1(x_1) + g(x_1 - x_2)]/m_1 = [g_2(x_2) - g(x_1 - x_2)]/m_2 \ . \tag{E}$$

If the two masses can be made to move in synchronism, i.e., if

$$g_1(x)/m_1 = g_2(x)/m_2$$

in some interval containing the origin, then (E) is possible with nonzero x_1 and $x_2 \equiv x_1$. Under such conditions, system (C), (D) is not distinguishable and the trivial solution of (C) is not asymptotically stable.

Conversely, if system (C), (D) is not distinguishable, then $\dot{x}_1 \equiv \dot{x}_2$ and (E) is true. Thus $x_1 - x_2 = c$ is constant. Substitute $x_1 = x + c$, $x_2 = x$ and $x = 0$ into (E). If $c \neq 0$, then one side of (E) is positive and the other side is negative. Thus, c must be zero and $g_1(x)/m_1 = g_2(x)/m_2$ for all x in some closed interval I which contains the origin in its interior. If there is no such interval I, then in this case (Case 4) the system (C), (D) is distinguishable and the trivial solution of system (C) is asymptotically stable in the large.

V. CONCLUDING REMARKS

We re-emphasize that in the present results, the asymptotic stability of the equilibrium of a system is ascertained by (a) determining the stability of the equilibrium of the corresponding conservative system, and (b) determining the attractivity of the equilibrium by examining the topological properties (i.e., an observability condition) of the entire damped system. Step (a) is easily verified for linear as well as for nonlinear systems. Step (b) is also easily verified for linear systems and for certain classes of nonlinear systems (e.g., nearly linear systems); however, in general, the verification of Step (b) for nonlinear systems may be quite difficult.

In any event, the present results provide added insight into the mechanisms of stability (and stabilization) for a large class of dynamical systems.

ACKNOWLEDGMENT

This work was supported by the National Science Foundation under Grant ENG 77-28446 and by the Engineering Research Institute, Iowa State University.

REFERENCES

[1] E. A. Barbasin and N. N. Krasovskii, "Stability of Motion in the Large," DAN 86 (1952), p.3.

[2] H. Goldstein, Classical Mechanics, Addison-Wesley, Reading, Mass., 1950.

[3] W. Hahn, Stability of Motion, Springer-Verlag, New York, 1967.

[4] J. P. LaSalle, "Asymptotic Stability Criterion," Proc. Symposia in Applied Mathematics, Vol. 13, American Math. Soc., Providence, R.I., 1962, pp. 299-307.

[5] E. B. Lee and L. Markus, Foundations of Optimal Control Theory, John Wiley and Sons, New York, 1967.

[6] J. J. Levin, "On the Global Asymptotic Behavior of Nonlinear Systems of Differential Equations," Arch. Rat. Mech. Anal. 6 (1960), pp. 65-74.

[7] R. K. Miller and A. N. Michel, "On Stabilization of Conservative Dynamical Systems", Proceedings of the Sixteenth Annual Allerton Conference on Communication, Control and Computing, University of Illinois, Urbana, Illinois, October 1978, pp. 557-565.

[8] P. P. Varaiya and R. Liu, "Normal Form and Stability of a Class of Coupled Nonlinear Networks," IEEE Transactions on Circuit Theory, Vol. CT-13(1966), pp. 413-418.

[9] J. A. Walker and W. E. Schmitendorf, "A Simple Test for Asymptotic Stability in Partially Dissipative Symmetric Systems," Trans. ASME, 95(1973), pp. 1120-1121.

[10] E. T. Whittaker, A Treatise on Analytical Dynamics of Particles and Rigid Bodies, Cambridge Univ. Press, 4th Edition, 1970.

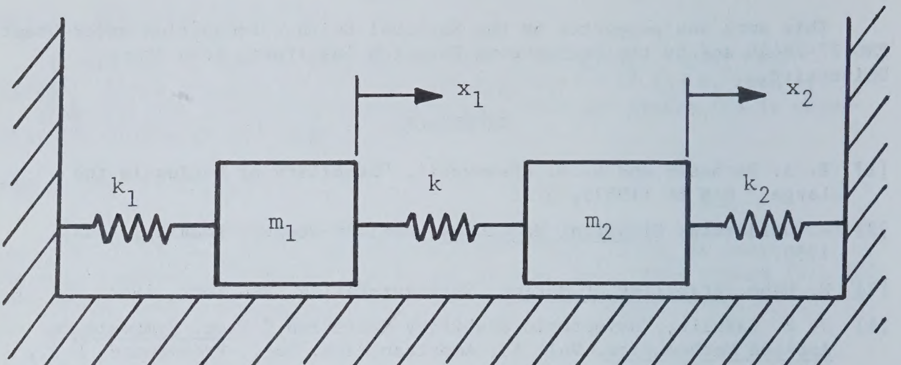

x1
x2
k1
m1
k
m2
k2

Fig. 1

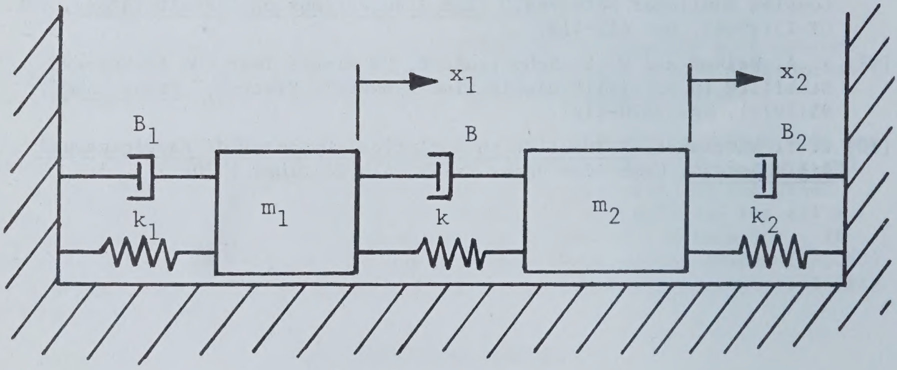

x1
x2
B1
B
B2
k1
m1
k
m2
k2

Fig. 2

ON A CLASS OF NONLINEAR ROBUSTNESS PROBLEMS[†]

B. Ross Barmish,[††] Lawrence E. Blume[†††] and Shirish D. Chikte[††]

Abstract: In this paper, new results are presented generalizing the existing "robustness theory" [1]-[3] for linear systems. With our more general formulation, we can accommodate nonlinear coupling between the uncertainty and the input. Analogous to the linear theory, we obtain a finite-dimensional necessary and sufficient condition for robustness. This test for robustness involves the unit simplex and the given data. An interesting special case is analyzed explicitly, illustrating the above results.

I. INTRODUCTION

In [1]-[3] a new notion of "robustness" is introduced for a class of perturbed linear dynamical equations. The systems under consideration are described in terms of an impulse response matrix whose entries are known only to lie within given bounds. A linear system, according to this notion, is said to be robust if an admissible input can be found such that the resulting output lies in a desired "target" set at a specified time instant, despite all possible variations in the impulse response matrix.

In this paper, we present some new results, whereby the existing theory is generalized so as to accommodate nonlinear coupling between uncertainties and input. Such a theory will enable us to handle problems involving robustness under "model uncertainty" mentioned above, as well as problems of target-set reachability of linear control systems perturbed by "additive noise" (see e.g. [4]-[7]).

The next section contains the basic assumptions about the perturbed dynamical system and the concept of robustness for such a system. The third section presents the main results; viz., (i) a necessary and sufficient condition for robustness which requires search over a finite-dimensional parameter space in lieu of function space, and (ii) a recipe helpful in computing a robust solution. The final section illustrates applications of these results by means of special cases and examples.

II. BASIC ASSUMPTIONS AND PROBLEM FORMULATION

The dynamical system under consideration will be defined by an integral flow of the following form

$$y(T) = \int_0^T A(x(t),q(t),t)dt \tag{2.1}$$

where

(i) $x(t) \varepsilon X \subset R^m$ is an m-dimensional solution (input) vector which is a function of time t;

[†] This research is supported by the U.S. Department of Energy under Contract Number ET-78-S-01-3390.

[††] Department of Electrical Engineering, University of Rochester, Rochester, New York 14627.

[†††] Department of Economics, University of Michigan, Ann Arbor, Michigan 48109.

(ii) $y(T) \varepsilon R^n$ is the *dependent* (output) vector at some future time $T > 0$, as a result of the solution $x(\cdot)$;

(iii) $q(t) \varepsilon Q \subset R^k$ is a k-vector of time-varying *uncertain* parameters reflecting our lack of full knowledge about the input-output relationship, and as such, $q(\cdot)$ is not available for measurement, and

(iv) $A: R^m \times R^k \times [0,T] \rightarrow R^n$ is the so-called *system function*, allowed to be an arbitrary nonlinear function, except for some mild regularity conditions which will be explained in the sequel.

We next introduce the *desired solution set* B which is described by a system of linear inequalities, i.e.,

$$B \overset{\Delta}{=} \{y \varepsilon R^n | a_i' y \le b_i \quad \text{for} \quad i = 1,2,\dots,r\} \tag{2.2}$$

where $a_i \varepsilon R^n$ and $b_i \varepsilon R$ are prespecified. We regard B as the "target" set within which the dependent variable $y(T)$ is required to lie *regardless* of the behavior of uncertain parameters $q(t)$ over $[0,T]$ (bounded by Q, of course).

We take the above five objects (T,X,Q,A,B) as prespecified and refer to them collectively as our *perturbed dynamical system* (S_p). Roughly speaking, "robustness" of (S_p) refers to the existence of a solution (input) function $x(\cdot)$ which results in a terminal output $y(T)$ that lies in the target set B for *all possible* uncertainty functions $q(\cdot)$. In order to make this notion precise, we let $M(X)$ and $M(Q)$ respectively denote the class of Lebesgue measurable functions on $[0,T]$ with values in X and Q almost everywhere (a.e.), and make the following definition.

Definition 1: An input (control) function $\tilde{x}(\cdot) \varepsilon M(x)$ is said to be a *robust solution* of (S_p) if $\int_0^T A(\tilde{x}(t),q(t),t)dt \varepsilon B$ *for all* $q(\cdot) \varepsilon M(Q)$. When such an $\tilde{x}(\cdot)$ exists, we say that (S_p) is *robust*.

Notice that if $Q = \{q_0\}$ and $B = \{b\}$ are *singleton* sets and $A(x,q,t) = A(q,t)x$ is *linear* in the *first* argument, the notion of robustness reduces to that of existence of a solution to a standard $\int_0^T A(q_0,t)x(t)dt = b$ linear integral equation. This analogy motivates the somewhat unconventional notation we have employed, viz. $x(\cdot)$ instead of $u(\cdot)$ for controls and A instead of H for the system function.

The main questions we are addressing here are: (i) can we develop a *computationally feasible necessary* and *sufficient condition for robustness*? and (ii) provided (S_p) *is* robust, can we prescribe a recipe for computing a robust control? Before we answer these questions, we need the following regularity assumptions on (S_p).

A0. The sets X and Q are non-empty and compact.

A1. For each fixed pair $x(\cdot) \varepsilon M(X)$ and $q(\cdot) \varepsilon M(Q)$, each component $A_i(x(t),q(t),t)$ is Lebesgue measurable and integrable over $[0,T]$.

A2. For each fixed pair $(x,t)\varepsilon X\times[0,T]$, $A(x,q,t)$ is continuous with respect to $q\varepsilon Q$.

A3. For each fixed pair $(q,t)\varepsilon Q\times[0,T]$, $A(x,q,t)$ is continuous with respect to $x\varepsilon X$.

It is clear that A0-A3 encompass a much wider class of systems than did the linear theory of [1]-[3]. Thus, we have not only allowed nonlinear coupling between input and uncertainty but also dependency among uncertainties entering in different components of the system function $A = (A_1, A_2, \ldots, A_n)'$. Furthermore, discontinuous changes in the system function over _time_ are _not_ ruled out .

III. MAIN RESULTS

Our main result tells us that we can decide whether a given system is robust by searching over the _finite-dimensional_ unit simplex

$$S \overset{\Delta}{=} \{z\varepsilon R^r: z \geq 0, \sum_{i=1}^{r} z_i = 1\} \tag{3.1}$$

in lieu of the function space $M(X)$. In order to state our main result, we must define the functions $h: R^r \to R$, $H: R^r\times[0,T] \to R$ and $\tilde{H}: R^m\times R^r\times[0,T] \to R$ by

$$h(z) \overset{\Delta}{=} -\sum_{i=1}^{r} z_i b_i \;; \tag{3.2}$$

$$H(z,t) \overset{\Delta}{=} \inf\{\tilde{H}(x,z,t): x\varepsilon X\} \tag{3.3}$$

where

$$\tilde{H}(x,z,t) \overset{\Delta}{=} \sum_{i=1}^{r} z_i \sup\{a_i' A(x,q,t): q\varepsilon Q\}. \tag{3.4}$$

Having this notation, we define the _parameterized objective function_ $F_e: R^r \to R$ by

$$F_e(z) \overset{\Delta}{=} h(z) + \int_0^T H(z,t)dt \;. \tag{3.5}$$

We then have

Theorem 1: _(S_p) is robust if and only if_

$$\max\{F_e(z): z\varepsilon S\} \leq 0 \;. \tag{3.6}$$

The detailed proof of the above theorem (forthcoming in [8]) is rather lengthy, and hence we only outline the main ideas involved.

Step 1: Utilizing some results [9] on commuting the integration and maximum operations, we first establish that robustness is equivalent to the existence of an $x(\cdot)\varepsilon M(X)$ such that

$$\int_0^T \max\{a_i' A(x(t),q,t): q\varepsilon Q\}dt - b_i \leq 0 \tag{3.7}$$

for all $i = 1,2,\ldots,r$.

Step 2: We show that it is possible to reformulate condition (3.7). Consequently, we obtain an equivalent problem over the *unit simplex* (3.1). Having done this, we have converted the r inequalities in (3.7) into an equivalent *single inequality* for robustness, given by

$$\min\{\max\{\sum_{i=1}^{r} z_i[\int_0^T \max\{a_i' A(x(t),q,t): q\varepsilon Q\}dt - b_i]: z\varepsilon S\}: x(\cdot)\varepsilon M(X)\} \leq 0 \quad (3.8)$$

where $z \triangleq (z_1, z_2, \ldots, z_r)'$.

Step 3: Noting that (3.8) still represents a *function space* optimization problem, we are motivated to invoke some theorems (see e.g. [10] and [11]) on compactness and convexity of *set-valued* integrals. These results help establish a *saddle point* for the above min-max problem, thereby allowing first a commutation for the min and max operations in (3.8), and subsequently, a commutation of minimization and integration becomes possible by [9]. This leads to condition (3.6), which evidently is our sought after finite-dimensional (concave) programming problem. ☐

If (S_p) is found to be robust, the next problem of obvious interest is the *computation* of a robust solution **(input)**. The following theorem is a step in this direction. The proof (also forthcoming in [8]) depends on saddle-point arguments.

Theorem 2: *There exists a pair* $(x_0(\cdot), z_0)\varepsilon M(X)\times S$, *such that* $x_0(\cdot)$ *is a solution to problem* (3.8), z_0 *is a solution to problem* (3.6) *and*

$$x_0(t)\varepsilon \arg\min\{\tilde{H}(x, z_0, t): x\varepsilon X\} \quad \text{a.e.} \quad (3.9)$$

or *equivalently*,

$$H(z_0, t) = \tilde{H}(x_0(t), z_0, t) \quad \text{a.e.} \quad (3.10)$$

Remark: The above theorem gives a useful *necessary* condition on robust solutions of (S_p) (whenever (S_p) is indeed robust), which in effect constitutes a pointwise minimum principle. Oftentimes, it indeed yields a robust control especially when the set of saddle-points is a *singleton*.

IV. ILLUSTRATIVE SPECIAL CASE

In this section, we illustrate the application of the preceding theory to the study of robustness properties of the class of *linear state equations* with *input coupling uncertainty*. Specifically, we consider the following model:

$$\dot{z}(t) = A(t)y(t) + [b(t) + q(t)]x(t), \; z(0) = 0. \quad (4.1)$$

where, $z(t)\varepsilon R^n$ is the state vector, $x(t)$ is a scalar input, $A(\cdot)$ is a (continuous) $n\times n$ matrix time function, $b(\cdot)$ is a bounded measurable n-vector time function and $q(\cdot)$ is the n-dimensional uncertainty vector. The uncertainty restraint set Q is described componentwise by $|q_i| \leq \overline{q_i}$, where $\overline{q_i} \geq 0$ is a prespecified bound on q_i, the i-th component of q. The control restraint set X will be simply $[-M,M]$ where $M > 0$ is given. Finally, the target set B will be assumed to be a rectangle given by $[\underline{b}_1, \overline{b}_1]\times[\underline{b}_2, \overline{b}_2]\times\ldots\times[\underline{b}_n, \overline{b}_n]$.

Under these conditions, it is well-known (cf. e.g. pp. 65-66 of [12]) that the state equation (4.1) has a unique solution for each $x(\cdot)\varepsilon M(X)$ and $q(\cdot)\varepsilon M(Q)$ which can be written as

$$y(t) = \int_0^t \phi(t,\tau)[b(\tau) + q(\tau)]x(\tau)d\tau \qquad (4.2)$$

where, $\phi(t,\tau) = [\phi_{ij}(t,\tau)]_{i,j=1,2,\ldots,n}$ is the so-called state-transition matrix associated with $A(\cdot)$. With T as the terminal time of interest, the system function $A\colon R\times R^n\times[0,T] \to R^n$ can now be identified as

$$A(x,q,t) = \phi(T,t)[b(t) + q]x \qquad (4.3)$$

We shall presently see that the various functions introduced in the last section take on fairly simple explicit forms for the class of problems above. Using the notation of Section II, we let $r = 2n$, $b_i = \overline{b}_i$ for $1 \le i \le n$, $b_i = -\underline{b}_i$ for $n+1 \le i \le 2n$; a_i is taken to be e_i, the unit vector in the i^{th} direction for $1 \le i \le n$ and we take $a_i = -e_{i-n}$ for $n+1 \le i \le 2n$. We shall furthermore write $\overline{z} = (z_1,z_2,\ldots,z_n)'$, $\underline{z} = (z_{n+1},z_{n+2},\ldots,z_{2n})'$, $\overline{b} = (b_1,b_2,\ldots,b_n)'$ and $\underline{b} = (b_{n+1},b_{n+2},\ldots,b_{2n})'$. Substitution in formulas (3.2)-(3.4) now yields

$$h(z) = \underline{z}'\underline{b} - \overline{z}'\overline{b}; \qquad (4.4)$$

$$\tilde{H}(x,z,t) = (\overline{z}-\underline{z})'A^o(t)x(t) + (\overline{z}+\underline{z})'\delta A(t)\ x(t)\ ; \qquad (4.5)$$

and

$$H(z,t) = M \min\{0,[(\overline{z}+\underline{z})'\delta A(t) - |(\overline{z}-\underline{z})A^o(t)||]\} \qquad (4.6)$$

where A^o and $\delta A(t)$ are n-vectors by

$$A^o(t) = \phi(T,t)b(t)$$

and

$$\delta A\,(t) = |\phi(T,t)|\overline{q};\quad \overline{q} \triangleq (\overline{q}_1,\overline{q}_2,\ldots,\overline{q}_n)'.$$

Above, we interpret the absolute value of a matrix entry by entry. The matrices $A^o(t)$ and $\delta A(t)$ may be interpreted respectively as the nominal impulse response vector and the vector of maximal allowable perturbations on the nominal impulse response. To illustrate the spirit of the above computations, we consider the double integrator whose input matrix is $b = [0,1]'$ and whose state transition matrix is

$$\phi(T,t) = \begin{bmatrix} 1 & (T-t) \\ 0 & 1 \end{bmatrix}.$$

It can be easily checked that the preceding expressions specialize for this example as follows:

$$A^o(t) = \begin{bmatrix} (T-t) \\ 1 \end{bmatrix};$$

$$A(t) = \begin{bmatrix} \bar{q}_1 + (T-t)\bar{q}_2 \\ \bar{q}_2 \end{bmatrix} ;$$

$$h(z) = -z_1\bar{b}_1 - z_2\bar{b}_2 + z_3\underline{b}_3 + z_4\underline{b}_4;$$

$$\tilde{H}(x,z,t) = [(z_1-z_3)(T-t) + (z_2-z_4)]x(t);$$

$$+ \{(z_1+z_3)\bar{q}_1 + [(z_1+z_3)(T-t) + (z_2+z_4)]\bar{q}_2\}|x(t)|$$

$$H(z,t) = M \cdot \min\{0, \bar{q}_2[(z_1+z_3)(T-t) + (z_2+z_4)]$$

$$+ (z_1+z_3)\bar{q}_1 - |(z_1-z_3)(T-t) + (z_2-z_4)|\}.$$

Remarks: We observe that the "candidate" robust control suggested by Theorem 2 turns out in this class of problems to be of the "bang-off-bang" type. The preceding formulas are easily seen to generalize in a straightforward fashion when one is dealing with the multi-input case.

The above example also brings out the fact that $F_e(z)$ is typically non-differentiable in z. Consequently, standard gradient methods cannot be used towards the computation of the maximum of $F_e(z)$. At present, we are in the process of developing a digital computer program that computes the maximum of $F_e(z)$ using the cutting hyperplane method (see [13] and [14]). This obviates the need to compute gradients. A detailed account of our computational experience will be forthcoming in [8].

REFERENCES

[1] B.R. Barmish, "Robust Solution of Perturbed Dynamical Equations from Within a Convex Restraint Set," IEEE Transactions on Auto. Control, in press.

[2] _____, "Robust Solution of Perturbed Linear Equations," IEEE Transactions on Auto. Control. AC-21, no. 7, pp. 123-124, 1977.

[3] B.R. Barmish and Y.H. Lin, "A New Notion of Robustness for Perturbed Linear Dynamical Systems," Proc. of 7th IFAC World Congress, Helsinki, 1978.

[4] M.C. Delfour and S.K. Mitter, "Reachability of Perturbed Systems and Min-sup Problems," SIAM J. Control, Vol. 7, pp. 521-533, 1969.

[5] D.P. Bertsekas and I.B. Rhodes, "On the Minimax Reachability of Target Sets and Target Tubes," Automatica, Vol. 7, pp. 233-247, 1971.

[6] H.S. Witsenhausen, "A Minimax Control Problem for Sampled Linear Systems," IEEE Trans. Auto. Control, AC-13, pp. 5-21, 1968.

[7] B.R. Barmish, "A Reachability Theory for a Class of Uncertain Dynamical Systems," Proc. 14th Allerton Conf. on Circuits and Systems, pp. 644-651, Monticello, Illinois, 1976.

[8] B.R. Barmish, L.E. Blume and S.D. Chikte, "Robustness of Systems with Arbitrary Input Uncertainties," in preparation.

[9] B.R. Barmish, "Measurable Selection Theorems and Their Application to Problems of Guaranteed Performance," IEEE Trans. Auto. Control, AC-23 pp. 685-687, 1978.

[10] R.J. Aumann, "Integrals of Set-Valued Functions," Jour. of Math. Analysis and Appl., Vol. 12, pp. 1-12, 1965.

[11] C. Olech, "Extremal Solutions of a Control Systems," Jour. Diff. Equations, Vol. 2, pp. 74-101, 1966.

[12] E.B. Lee and L. Markus, Foundations of Optimal Control Theory, John Wiley, New York, 1967.

[13] A.F. Veinott, Jr., "The Supporting Hyperplane Method for Unimodal Programming," Operations Research, Vol. XV, No. 1, pp. 147-152, 1967.

[14] J.E. Kelley, Jr., "The Cutting-Plane Method for Solving Convex Programs," SIAM, Vol. 8, pp. 703-712, 1960.

ESTIMATION AND CONTROL OF NONLINEAR SYSTEMS HAVING CONE-BOUNDED NONLINEARITIES

B.M. SCHERZINGER and R.H.S. KWONG
Department of Electrical Engineering
University of Toronto
Toronto, Ontario, Canada M5S 1A4

ABSTRACT

This paper addresses the problems of suboptimal regulation and control for a class of stochastic systems whose nonlinear plant and observation dynamics are cone-bounded. The problems are considered on finite and infinite time intervals, and sufficient conditions are presented to ensure estimator and closed loop stability in the form of stability conditions for upper bounds on the estimation error- and closed-loop state covariances.

1. INTRODUCTION

The general problems of estimation and control of nonlinear stochastic systems remain to date unsolved, and as a result, alternative approaches are being considered in the literature, leading to suboptimal yet acceptable solutions based on alternate design criteria such as simplicity, robustness and ease of implementation into an applications setting (see for example Safanov and Athans [5]). Among these are the results of Gilman and Rhodes [1], [2], which deal with the estimation and regulation of a subclass of systems having nonlinear plant and observation functions which satisfy conditions of incremental and absolute conicity. They applied the intuitively appealing "model reference estimator" to the system and used information describing the conicity to obtain an upper bound on the mean-square performance of the estimator. Applying a linear control law to the system, they established a similar bound on the incurred quadratic cost.

The present work is an extension of these results. The system model now contains state-dependent as well as additive noise processes. Also, the resulting upper bounds are shown to have stability properties under suitable sufficient conditions which are primarily restrictions on the degree of nonlinearity in the system and the intensity of the state-dependent noise. In this way conditions are obtained which are sufficient for the stability of the estimator and/or the regulated system.

The system to be considered is

$$dx_t = f(x_t,u_t,t)dt + dG_t x_t + d\nu_t \tag{1a}$$

$$dz_t = g(x_t,t)dt + d\omega_t \tag{1b}$$

ν_t and ω_t are independent Wiener noise processes, appropriately dimensioned, with

$$E[d\nu_t d\nu_t^T] = V(t)dt \qquad \text{and} \qquad E[d\omega_t d\omega_t^T] = W(t)dt$$

G_t is a square matrix of Wiener noise processes independent of ν_t and ω_t. An alternate representation of the state-dependent noise is

$$dG_t x_t = Y(x_t,t)d\eta_t \quad \text{where} \quad d\eta_t = \mathrm{col}[dG_t]$$
$$\text{and} \quad E[d\eta_t d\eta_t^T] = \Xi(t)dt$$

The nonlinear functions f and g are uniformly Lipschitz in x_t and u_t, and satisfy the incremental conicity condition

$$\|f(x_t+\epsilon,u_t+\delta,t)-f(x_t,u_t,t)-A(t)\epsilon-B(t)\delta\| \leq a(t)\|\epsilon\|+b(t)\|\delta\| \tag{2a}$$

$$\|g(x_t+\epsilon,t)-g(x_t,t)-C(t)\epsilon\| \leq c(t)\|\epsilon\| \tag{2b}$$

and the condition of absolute conicity

$$\|f(x_t,0,t) - H(t)x_t\| \leq h(t)\|x_t\| \tag{2c}$$

The cone-bounding parameters a(t),b(t),c(t) and h(t) are non-negative scalars which quantify the nonlinear deviation of (1a,b) from the following nominal linear system

$$dx_t = A(t)x_t dt + B(t)u_t dt + dG_t x_t + d\nu_t \tag{3a}$$

$$dz_t = C(t)x_t dt + d\omega_t \tag{3b}$$

For this system (3a,b), Gustafson and Speyer [3] presented the linear minimum variance (LMV) filter, given by

$$d\hat{x}_t = A(t)\hat{x}_t dt + \overline{K}(t)[dz_t-C(t)\hat{x}_t dt], \quad \hat{x}_0=E[x_0] \tag{4a}$$

where

$$\begin{aligned}\overline{K}(t) &= \overline{P}(t)C(t)^T W(t)^{-1}\\ \overline{P}(t) &= E[(x_t-\hat{x}_t)(x_t-\hat{x}_t)^T]\end{aligned} \tag{4b}$$

$$\begin{aligned}\frac{d\overline{P}}{dt} &= V + A\overline{P} + \overline{P}A^T - \overline{P}C^T W^{-1} C\overline{P} + \Delta[\overline{X},t]\\ \overline{P}(0) &= \mathrm{cov}[x_0]\end{aligned} \tag{4c}$$

where

$$\overline{X}(t)=E[x_t x_t^T], \quad \Delta[\overline{X}(t),t]dt=E[dG_t\overline{X}(t)dG_t^T] \tag{4d}$$

and $$\frac{d\overline{X}}{dt} = V + A\overline{X} + \overline{X}A^T + \Delta[\overline{X},t], \quad \overline{X}(0) = \mathrm{cov}[x_0] \tag{4c}$$

This estimator is optimal among linear estimators though not among the general class of causal estimators, as application of the optimal nonlinear filtering equations [4] to (3a,b) will result in an estimator of infinite order.

The following terminology is used in this paper. Matrices are written with capital letters; stochastic processes are written with subscripted letters. The columnization of a matrix A is written col[A]. For purposes of brevity, time arguments of functions will often be suppressed. Also, all proofs of results are omitted, and can be found in [6].

2. ESTIMATION - MAIN RESULTS

For the estimation problem, we assume $u_t=0$, although in principle a deterministic control input can be lumped with the time dependence of f. We use the suboptimal model-reference estimator

$$d\hat{x}_t = f(\hat{x}_t,0,t)dt + K(t)[dz_t-g(\hat{x}_t,t)dt], \quad \hat{x}_0 = E[x_0] \tag{5}$$

The estimator gain K(t) parametrizes a class of upper bounds on the estimator mean-square performance (error covariance) as follows:

$$E[(x_t-\hat{x}_t)(x_t-\hat{x}_t^T] \leq \tilde{P}(t)$$

where $\tilde{P}(t)$ is the solution of

$$\frac{d\tilde{P}}{dt} = [A-KC]\tilde{P}+\tilde{P}[A-KC]^T+V+K[W+\Gamma[\tilde{P}]]K^T+\pi[\tilde{P}]+\Delta[X,t]$$

$$\tilde{P}(0) = \mathrm{cov}[x_0] \tag{6}$$

where

$$\pi[P(t)]=[a(t)+c(t)]P(t)+a(t)\mathrm{tr}[P(t)]I, \quad \Gamma[P(t)]=c(t)\mathrm{tr}[P(t)]I$$

Here, $X(t)$ is the solution of

$$\frac{dX}{dt} = V + HX + XH^T + \xi[X,t], \qquad X(0) = \mathrm{cov}[x_0] \tag{7}$$

with

$$\xi[X(t),t] = h(t)[X(t)+\mathrm{tr}[X(t)]I] + \Delta[X(t),t]$$

It is an upper bound on the state covariance, i.e.,

$$E[x_t x_t^T] \le X(t)$$

In this class of bounds, there exists a matrix-ordering minimum bound $P(t)$, defined by

$$\frac{dP}{dt} = \hat{V} + AP + PA^T - PC^T\hat{W}^{-1}CP + \pi[P]$$

$$P(0) = \mathrm{cov}[x_0] \tag{8}$$

where

$$\hat{V}(t) = V(t) + \Delta[X(t),t], \qquad \hat{W}(t) = W(t) + \Gamma[P(t)]$$

We achieve the minimum $P(t)$ by choice of the following estimator gain

$$K(t) = K_0(t) = P(t)C(t)^T\hat{W}(t)^{-1} \tag{9}$$

The following theorems summarize these results.

Theorem 2.1: If the estimator (5) is applied to the system (1a,b), whose nonlinear functions satisfy a uniform Lipschitz condition in x_t and the conicity conditions (2a,b,c), then the state-covariance is bounded above by $X(t)$ satisfying (7), and the estimator error covariance is bounded above by $\tilde{P}(t)$ satisfying (6).

Proof: The development is sufficiently similar to that of [1] and is not repeated here. The only difference is the appearance of the term $\Delta[X(t),t]$ which occurs as a result of the state-dependent noise.

Theorem 2.2: Let $P(t)$ and $\tilde{P}(t)$ be the solutions of (8) and (6) respectively. Then for $t\ge0$, $P(t)\le\tilde{P}(t)$.

This is the well-known minimum property of matrix Riccati equations. An adequate proof is available in [7]. On the infinite time interval, we define stability of the estimator to be that of the stability of its error covariance. This in turn is ensured by the stability of its upper bound. The term $\Delta[X(t),t]$ is bounded on $t\epsilon[0,\infty)$ if $X(t)$ is also bounded, and therefore the stability of $X(t)$ will usually be required as on of several sufficient conditions for estimator stability.

The Riccati equation structure of (8) admits the use of results of Wonham [7], [8], dealing with the boundedness and convergence of solutions of matrix Riccati equations.

Theorem 2.3: (7) has a solution $X(t)$ such that

(i) $X(t)$ is defined, absolutely continuous and unique on $t\geq 0$.

(ii) If the parameters H, h and V are constant, if H is stable, if Δ is independent of t, and

$$\left\|\int_0^\infty e^{Ht}[E[dG_t dG_t^T] + (n+1)he^{H^T t}dt\right\| < 1 \tag{10}$$

then $X_\infty = \lim_{t\to\infty} X(t)$

exists and is the unique positive semidefinite solution of

$$V + HX_\infty + X_\infty H^T + \Delta[X_\infty] + hX_\infty + h\,\mathrm{tr}[X_\infty]I = 0 \tag{11}$$

Theorem 2.4: Assume that $\Delta[X(t),t]$ is bounded on $t\in[0,\infty)$. Then (8) has a solution $P(t)$ such that

(i) $P(t)$ is defined and absolutely continuous for $t\geq 0$. Also, $P(t)\geq 0$ and is the unique solution.

(ii) If the parameters A,C,H,W,V,a,h and c are constants, if (C,A) is detectable, if Δ is independent of t, and

$$\inf_K \left\|\int_0^\infty e^{t(A-KC)}[(n+1)aI + (nKK^T+I)c]e^{t(A-KC)^T}dt\right\| < 1 \tag{12}$$

then $P(t)$ is bounded on $t\in[0,\infty)$.

(iii) If in addition $\hat{V}(t)\to V_\infty$ (that is, $\Delta[X(t)]$ approaches a stationary value) such that $V_\infty = MM^T$ and (A,M) is controllable, then

$P_\infty = \lim_{t\to\infty} P(t)$

exists and is the unique positive semidefinite solution of the algebraic Riccati equation

$$V_\infty + AP_\infty + P_\infty A^T - P_\infty C^T\hat{W}^{-1}CP_\infty + \pi[P_\infty] = 0 \tag{13}$$

Furthermore, the matrix $A-C^T\hat{W}^{-1}CP_\infty$ is stable.

These results provide a simple and constructive approach to the design problem of stable estimators for cone-bounded nonlinear stochastic systems. Equation (8) for the error covariance upper bound is structurally similar to the equation describing the error covariance of the Kalman-Bucy filter for the system

$$dx_t = A(t)x_t dt + d\nu_t$$

$$dz_t = C(t)x_t dt + d\omega_t$$

The added conditions (10) and (12) place limits on the degree of nonlinearity and the amount of state-dependent noise which the system can tolerate. Two one-point boundary value problems in symmetric matrices need to be solved in order to determine the bound-minimizing estimator. In this respect, the complexity of the calculations is no worse than is required for the LMV estimator. Moreover, the LMV estimator results with vanishing nonlinearity and hence is continuously embedded in the nonlinear case. Define as a measure of system nonlinearity

$$d_E = \sup_{t\geq 0} \{a(t), b(t), c(t)\}$$

Theorem 2.5: Let $\tilde{P}(t)$ and $\overline{P}(t)$ be the solutions of (6) and (4c) respectively, where in (6),

$$K(t) = \overline{P}(t)C(t)^T[W(t)+\Gamma[\overline{P}(t)]]^{-1} \tag{14}$$

Then $\overline{P}(t)\leq\tilde{P}(t)$, and $\tilde{P}(t)-\overline{P}(t)$ vanishes t-uniformly on the order of d_E with vanishing d_E.

We have a complete circle of inclusion in that with vanishing state-dependent noise, our filter converges to that described in Gilman and Rhodes [1], which in turn converges with vanishing d_E to a K-B filter. Such inclusions are desirable, as we can effectively deal with nonlinear systems which are "nearly linear", and hence the estimator is also "nearly linear".

As a final note, we present an alternative approach to the evaluation of conditions (10) and (12). In (12) for example, the integral is equivalent to the solution of

$$(A-KC)\hat{P} + \hat{P}(A-KC)^T + \pi[I] + K\Gamma[I]K^T = 0$$

The problem reduces to finding a suitable K such that

$$\inf_K ||\hat{P}|| = \inf_K \lambda_{max}[\hat{P}] < 1$$

where λ_{max} denotes the maximum eigenvalue.

3. REGULATION - MAIN RESULTS

The linear quadratic (LQ) and linear quadratic Gaussian (LQG) regulator problems have known optimal solutions exhibiting several desirable features, such as linearity of the optimal control, and in the LQG case, separation of estimation and control design. When nonlinear systems are included, these properties hold, and, as in the estimation problem, suboptimal design schemes with attractive secondary features become of interest. In this context, the linear control law applied to a cone-bounded nonlinear system is an appealing choice, as it admits the derivation of an upper bound on the quadratic cost incurred over a finite time interval. The cost function is

$$J_t = x_T^T S_T x_T + \int_t^T [u_s^T R(s) u_s + x_s^T Q(s) x_s] ds \tag{15}$$

The control law to be used is

$$u_t = L(t)\hat{x}_t \tag{16}$$

where $\hat{x}_t$ is a causal estimate provided by the estimator (5). The stochastic regulation problem for incomplete state information is considered in detail here, and the problem with complete state information is considered as a special case, obtained by setting $\hat{x}_t=x_t$ and hence $P(t)=0$.

Theorem 3.1: Let the control law (16) be applied to the system (1a,b), where the estimate of the state derives from the causal estimator (5) having a mean-square performance upper-bounded by P(t). Then the expectation of the quadratic cost on the time interval [t,T] is upper-bounded as follows.

$$E\{J_t\} \leq E\{x_t^T S(t) x_t\} + \int_t^T tr[S(s)V(s)+M(s)P(s)]ds \tag{17}$$

where S(t) is the solution of

$$-\frac{dS}{dt} = Q+L^TRL+S[A+BL]+[A+BL]^TS+[a+b]S+a\ tr[S]I+b\ tr[S]L^TL+\psi[S,t]+nI$$

$$S(T) = S_T \tag{18}$$

where

$$n(t) = ||N(t)||$$

$$N(t) = S(t)B(t)L(t) + L(t)^T[R(t)+b(t)tr[S(t)]I]L(t)$$

$$M(t) = L(t)^T[R(t)+b(t)tr[S(t)]I]L(t) + n(t)I$$

and ψ is defined by

$$tr[Y(x_t,t)\Xi(t)Y(x_t,t)^TS(t)] = x_t^T\psi[S(t),t]x_t$$

The linear feedback matrix L(t) parametrizes a class of upper bounds on the quadratic cost, among which the minimum is obtained by choice of

$$L(t) = L_0(t) = -[R(t)+b(t)tr[S(t)]I]^{-1}B(t)S(t) \tag{19}$$

The resulting equation describing S(t) is

$$-\frac{dS}{dt} = Q + A^TS + SA - SB\hat{R}[S]^{-1}B^TS + \psi[S,t] + \gamma[S]$$

$$S(T) = S_T \tag{20}$$

where

$$\hat{R}[S] = R + b\ tr[S]I, \quad \gamma[S] = (a+b)S + a\ tr[S]I$$

Theorem 3.2 (minimum property): Let $\tilde{S}(t)$ and S(t) be solutions of (18) and (20) respectively. Then for $t \le T$, $\tilde{S}(t) \ge S(t)$.

The Riccati structure of (20) permits exploitation of the previously mentioned results of Wonham for the infinite time stability of its solution.

Theorem 3.3: Let S(t) be the solution of (20). Then if the parameters A,B,Q,R,a and b are constant, if ψ is a linear positive, time invariant operator, if (A,B) is stabilizable, if $Q=D^TD$ where (D,A) is observable, and

$$\inf_L \left|\left| \int_0^\infty e^{(A+BL)^Tt}[(n+1)aI+(nL^TL+I)b+E[dG_tdG_t^T]]e^{(A+BL)t}dt \right|\right| < 1 \tag{21}$$

then

$$S_\infty = \lim_{t\to-\infty} S(t)$$

exists and is the unique positive-semidefinite solution of the algebraic Riccati equation

$$Q + A^TS_\infty + S_\infty A - S_\infty B\hat{R}[S_\infty]B^TS_\infty + \psi[S_\infty] + \gamma[S_\infty] = 0 \tag{22}$$

Furthermore, the matrix $A-B\hat{R}[S_\infty]^{-1}B^TS_\infty$ is stable.

(21) restricts the magnitudes of the cone-bounding parameters and the state-dependent noise intensity. These conditions are not however sufficient for the stability of the closed-loop system, consisting of the control law (16) and the estimator (5), since the estimator gain matrix K(t) depends on an upper bound on the closed-loop state covariance. It therefore becomes necessary to consider the state, and the estimate, or more conveniently the estimation error, as a coupled system, given as follows

$$y_t = \begin{bmatrix} x_t \\ \tilde{x}_t \end{bmatrix}$$

$$dy_t = k(y_t,t)dt + d\overline{G}_t + d\ {}_t \qquad (23)$$

where

$$k(y_t,t) = \begin{bmatrix} f(x_t, L(t)(x_t - x_t), t) \\ \tilde{f}_t - K(t)\tilde{g}_t \end{bmatrix}$$

$$d\overline{G}_t = \begin{bmatrix} dG_t & 0 \\ dG_t & 0 \end{bmatrix}, \qquad d\sigma_t = \begin{bmatrix} I & 0 \\ I & -K(t) \end{bmatrix}\begin{bmatrix} d\nu_t \\ d\omega_t \end{bmatrix}$$

Theorem 3.4: The state-covariance of the system (23) is upper-bounded as follows:

$$E[y_t y_t^T] \le Y(t) = \begin{bmatrix} Y_{11}(t) & Y_{12}(t) \\ Y_{21}(t) & Y_{22}(t) \end{bmatrix}$$

where Y(t) is the solution of

$$\frac{dY}{dt} = \overline{A}Y + Y\overline{A}^T + \overline{V} + \overline{\Delta}[Y,t] + [a+b+c]Y + a\,tr[Y]I + b\,tr[\overline{L}^T\overline{L}Y]I + c\,tr[M_{22}Y]\overline{K}\,\overline{K}^T \qquad (24)$$

$$Y(0) = \begin{bmatrix} cov[x_0] & 0 \\ 0 & cov[x_0] \end{bmatrix}, \qquad \overline{A} = \begin{bmatrix} A+BL & -BL \\ 0 & A-KC \end{bmatrix}$$

$$\overline{K} = \begin{bmatrix} 0 & 0 \\ 0 & K \end{bmatrix}, \qquad \overline{V} = \begin{bmatrix} V & V \\ V & V+KWK^T \end{bmatrix}$$

$$\overline{L} = \begin{bmatrix} L & -L \\ 0 & 0 \end{bmatrix}, \qquad M_{22} = \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix}$$

$$\overline{\Delta}[Y(t),t]dt \triangleq E[d\overline{G}_t Y(t) d\overline{G}_t^T]$$

Using this notation, we generate the inequalities

$$E[x_t x_t^T] \le Y_{11}(t) \triangleq X(t), \qquad E[\tilde{x}_t \tilde{x}_t^T] \le Y_{22}(t) \triangleq P(t)$$

The composite covariance Y(t) is used to generate sufficient conditions for the stability of the closed loop system, as given in the following theorem.

Theorem 3.5: Let Y(t) be the solution of (24). Then if $\overline{A},\overline{K},\overline{V},\overline{L}$,a,b and c are constant, if $\overline{A}$ is stable, if $\overline{\Delta}$ is independent of t, and

$$\left\|\int_0^\infty e^{\overline{A}t}[(2n+1)aI + (tr[\overline{L}^T\overline{L}+1)bI + (2n\overline{K}\,\overline{K}^T+I)c + E[dG_t dG_t^T]]e^{\overline{A}^T t}dt\right\| < 1 \qquad (25)$$

then

$$Y_\infty = \lim_{t\to\infty} Y(t)$$

exists and is the unique positive semidefinite solution of the algebraic equation

$$\bar{A}Y_\infty+Y_\infty\bar{A}^T+\bar{V}+[a+b+c]Y_\infty+a\,\mathrm{tr}[Y_\infty]I+b\,\mathrm{tr}[\bar{L}^T\bar{L}Y_\infty]I+c\,\mathrm{tr}[Y_\infty]\bar{K}\bar{K}^T+\bar{\Delta}[Y_\infty]=0 \qquad (26)$$

The results of this section combine to form a design procedure for stable controls for cone-bounded nonlinear systems. When the complete state is available, no estimator is required, and the results of Theorem 3.5 simply reduce to stability conditions on the state-covariance upper bound. As sufficient conditions for closed-loop stability, (21) guarantees a stationary stabilizing L_0, and then (25) ensures that the resulting closed-loop system is stable.

If only partial observations are available, then the estimator (5) together with the control (16) form a dynamic compensator which stabilizes the system when the stability conditions are met. We have not been able to separate the design problems of estimation and control in the presence of state-dependence noise, as in this case, coupling exists between the state- and error covariance upper bounds, as was shown in Section 2. A stationary stabilizing estimator gain K_0 is thus not separately obtainable. When no state-dependence noise is present, such a gain is then separately obtainable. In either case, the conditions of Theorem 3.5 are sufficient for closed-loop stability.

CONCLUSIONS

We have presented an alternative approach to the problems of estimation and control for the restricted class of cone-bounded nonlinear systems which include state-dependent noise. The approach has basically been to determine mean-square performance upper bounds on the estimator and the regulated system, and then define the stability of the estimator and/or the closed-loop system in terms of these bounds. The sufficient conditions ensuring stability include bounding conditions on the magnitudes of the cone-bounding parameters and the state-dependent noise intensity. The design procedure can be interpretted as being one of determining the amount of nonlinearity which an initially linear system can tolerate and still retain closed-loop stability. It should be noted that, with these results, no conclusions about the behaviour of the actual m.s. performances can be made.

REFERENCES

[1] A.S. Gilman and I.B. Rhodes, "Cone-bounded nonlinearities and mean-square bounds — estimation upper bounds", IEEE Trans. Auto. Control, vol. AC-18, no. 3, June 1973, pp. 260-265.

[2] A.S. Gilman and I.B. Rhodes, "Cone-bounded nonlinearities and mean-square bounds — quadratic regulation bounds", IEEE Trans. Auto. Control, vol. AC-21, no. 4, Aug. 1976, pp. 472-483.

[3] D.E. Gustafson and J.L. Speyer, "Linear minimum variance filters applied to carrier tracking", IEEE Trans. Auto. Control, vol. AC-21, no. 1, Feb. 1976, pp. 65-73.

[4] A.H. Jazwinski, Stochastic Processes and Filtering Theory, Academic Press, New York, 1970.

[5] M.G. Safanov and M. Athans, "Robustness and computational aspects of nonlinear stochastic estimators and regulators", IEEE Trans. Auto. Control, vol. AC-23, no. 4, Aug. 1978.

[6] B.M. Scherzinger, "Estimation and control of nonlinear systems having cone-bounded nonlinearities", M.A.Sc. Thesis, University of Toronto, Sept. 1979.

[7] W.M. Wonham, "On a matrix Riccati equation of stochastic control", SIAM J. of Control, vol. 6, no. 4, 1968, pp. 681-697.

[8] W.M. Wonham, Linear Multivariable Control, Springer-Verlag, New York, 1974.

DETERMINING THE FEASIBLE REGION FOR NON-STATIONARY NON-LINEAR FEEDBACK SYSTEMS

HEBERTT SIRA-RAMIREZ
Control Systems Department, Universidad de Los Andes, Mérida-Venezuela

ABSTRACT

This paper presents a Carleman's linearization based method for obtaining arbitrarily close approximations to the feasible region of non-linear time-varying feedback control systems. The set of feasible states is characterized as a Generalized Polyhedron [15] whose support vectors evolution equations are explicitely established. Examples and implications of the results are included.

I INTRODUCTION

The set of feasible states for dynamic unforced systems is defined as the set of all possible solutions of the systems differential equation, at certain prescribed instant of time, arising from every initial state confined within the bounds of a compact set in the state space of the system.

Thus far, the studies dealing with the feasible set have been restricted to the linear case (Witsenhausen [1], Podchukaev [2], Sira-Ramirez [3]) and as initial state sets only ellipsoids and polyhedra have been considered (Schweppe [4], Hnyilicza [5], Bertsekas [6]). As applications of the feasible set concept, authors ([4],[5],[6], Schlaepfer [7], Sira-Ramirez [8]) have defined estimation and control problems from a non-probabilistic framework whose importance in applications is the basis of currently on-going research (Barmish [9],[10],[11],[12], Schmittendorf [13], Chukwu [14]).

This work tries to lift the linear case, as well as the ellipsoids and polyhedra,restrictions on set-theoretic uncertainties evolution problems by proposing the use of Carleman's linearization technique for feasible set computation on a class of non-linear analytic feedback systems. The technique allows arbitrarily close approximations to the feasible region of the non-linear system by considering the feasible set of a high-dimensional linear system with sparse system matrix.

Section II presents some basic definitions related to tensor powers of n-dimensional vectors, families of tensor powers, their use in defining Generalized Polyhedra (GP) as well as some other basic results regarding associated tensor systems to basic linear systems. (See also Brockett [19],[20]) This section contains a basic theorem due to Loparo and Blankenship [18] which constitutes the essence of later results about feasible regions of non-linear analytic feedback systems.

Section III deals with the basic problem treated in this paper, namely to determine the feasible region of a non-linear analytic system whose initial state is characterized by a GP (See also [15], [16]). This problem is readily solved using Loparo's and Blankenship result. The evolution equations for the Generalized Support Vecotrs (GSV) defining the feasible set are obtained via standard linear techniques.

Section IV discusses the implications of the results in set-theoretic estimation and control problems for non-linear systems. Some suggestions for further research are given at the end of this section.

II BASIC DEFINITIONS AND RESULTS

In this section we present some definitions which can also be found in [15],[16], and [19].

If $\underline{x}$ is an n-vector with components $x_1, x_2, \ldots x_n$ we denote $\underline{x}^{[p]}$ the $\binom{n+p-1}{p}$-dimensional vector of p-forms in $x_1, x_2, \ldots x_n$ (i.e the elements of of $\underline{x}^{[p]}$ are of the form : $\alpha \Pi_{i=1}^{m} x_i^{p_i}$ with $\Sigma p_i = p$: $p_i \geqq 0$ and

$$\alpha = \sqrt{\binom{p}{p_1}\binom{p-p_1}{p_2}\cdots\binom{p-p_1-p_2-\cdots-p_{n-1}}{p_n}}$$

If $\underline{y} = A\,\underline{x}$ then $\underline{y}^{[p]} = A^{[p]}\underline{x}^{[p]}$ is verified and $A_{[p]}$ denotes the infinitesimal version of $A^{[p]}$, i.e $d/dt\ \underline{x}(t) = A(t)\ \underline{x}(t)$ implies $d/dt\ \underline{x}^{[p]}(t) = A_{[p]}(t)\ \underline{x}^{[p]}(t)$. Some useful properties of these "tensor powers" are : 1) $(AB)^{[p]} = A^{[p]}B^{[p]}$; 2) $(A^q)^{[p]} = (A^{[p]})^q$ whenever A^q is defined and q is an integer. 3) $(A')^{[p]} = (A^{[p]})'$

Lemma 1 $d/dt\ \Phi_A^{[p]}(t,t_0) = A_{[p]}\Phi_A^{[p]}(t,t_0)$ whenever $\Phi_A(t,t_0)$ is the state transition matrix associated with $A(t)$.

A proof of this simple fact can be found in [15].

We denote $\underline{x}^{(\underline{p})}$ (notice the vector character of p) the $\binom{n+p}{p}$-dimensional vector $(\ \underline{x}', (\underline{x}^{[2]})', \ldots, (\ \underline{x}^{[p]})'\ )'$ (We also call this vector the p-th family power of $\underline{x}$). By extension of the above definitions if $\underline{y} = A\ \underline{x}$ then $\underline{y}^{(\underline{p})} = A^{(\underline{p})}\ \underline{x}^{(\underline{p})}$ where $A^{(\underline{p})}$ is a block-diagonal matrix of the form: $\operatorname{diag}[\ A\ ,\ A^{[2]}\ ,\ \ldots,\ A^{[p]}\ ]$. It is easy to see that if $d/dt\ \underline{x}(t) = A(t)\underline{x}$ then $d/dt\ \underline{x}^{(\underline{p})}(t) = A_{(\underline{p})}(t)\ \underline{x}^{(\underline{p})}(t)$ where $A_{(\underline{p})}(t)$ is the infinitesimal version of $A^{(\underline{p})}(t)$.

The following lemma extends the previous one to the case of p-th family powers of state transition matrices for linear systems.

Lemma 2 $d/dt\ \Phi_A^{(\underline{p})}(t,t_0) = A_{(\underline{p})}(t)\ \Phi_A^{(\underline{p})}(t,t_0)$

Definition 1 A Generalized Polyhedron (GP) is a set of the form:

$$\{\ \underline{x} \in R^n : \langle \underline{h}_i,\ \underline{x}^{(\underline{p})} \rangle \leqq 1\ ;\ i = 1,2,\ldots,M\ \}$$

$\underline{h}_i$ is called a generazlized support vector (GSV). The set $\{\underline{h}_i : i=1,2,\ldots,M\}$ is the generalized support set. Note that a GP may well represent a non-convex, not necessarily bounded set. Polyhedra, spheres, ellipsoids, zonoids, etc. may be viewed as particular cases of GP

Definition 2 Given the linear system: $d/dt\ \underline{x}(t) = A(t)\ \underline{x}(t)$ we denote by $\underline{\xi}(t,\underline{x}_0)$ the trajectory starting from $\underline{x}_0$. A state $\underline{x}(T)$ is said to be $\underline{x}_0$-feasible if $\underline{\xi}(T,\underline{x}_0) = \underline{x}(T)$ for a certain $T > 0$. Let Σ_0 be a closed set in R^n, we define the Σ_0-feasible set (region),for the given linear system, at time T the set : $\{\underline{x} \in R^n :\ \ \underline{x} = \ \underline{\xi}(T,\underline{x}_0)$ for some $\underline{x}_0 \in \Sigma_0\ \}$. We denote this set by $\underline{\xi}(T,\ \Sigma_0)$. The Σ_0-feasible set at time t is then the set of all $\underline{x}_0$-feasible states at time t for all $\underline{x}_0 \in \Sigma_0$

Proposition 1 The Σ_0-feasible set for the linear system $d/dt\ \underline{x} = A(t)\ \underline{x}$ is a GP characterized by the support vectors $\underline{h}_i(t)$; $i=1,2,\ldots,M$, whenever Σ_0 is a GP characterized by the GSV $\underline{h}_{i0}$. Moreover $\underline{h}_i(t)$ is given, for each i , by the unique solution at time t of the linear vector differential equation: $d/dt\ \underline{h}_i(t) = -\ A'_{(\underline{p})}(t)\ \underline{h}_i(t)$ with $\underline{h}_i(t_0) = \underline{h}_{i0}$

Proof The form of the Σ_0-feasible set follows from the affine classification theorem [10]. The computation of the GSV of $\underline{\xi}(t,\Sigma_0)$ is as follows : $\underline{\xi}(t,\Sigma_0) =' \{\underline{x} \varepsilon R^n : \underline{x} = \Phi_A(t,t_0)\underline{x}_0$ for all $\underline{x}_0 \varepsilon \Sigma_0\} =' \{\underline{x} \varepsilon R^n : \Phi_A(t_0,t)\underline{x} \varepsilon \Sigma_0\}$ $=' \{\underline{x} \varepsilon R^n : <(\Phi_A(t_0,t)\underline{x})^{\underline{(p)}}, \underline{h}_{i0}> \leqq 1$; for all i }. Letting $\underline{h}_i(t) = \Phi_A'^{\underline{(p)}}(t_0,t)\ \underline{h}_{i0}$ we have $\underline{\xi}(t,\Sigma_0) =' \{\underline{x} \varepsilon R^n : <\underline{x}^{\underline{(p)}}, \underline{h}_i(t)> \leqq 1$ for all i } The result follows from the definition of $\underline{h}_i(t)$, lemma 2 and well known properties of time derivatives of an inverse matrix.

Definition 3 Let $\underline{f}(\underline{x},t)$ be an R^n-valued uniformly bounded analytic function of $\underline{x}$ and piecewise continuous in t. Then we can express $f(\underline{x},t)$ as an infinite series in terms of the tensor powers of $\underline{x}$. Furthermore if $\underline{f}(0,t) = 0$ for all t, then the series takes the form $\underline{f}(\underline{x},t) = \sum_{k=1}^{\infty} F_k(t)\ \underline{x}^{[k]}$

Definition 4 Let $d/dt\ \underline{r}(t) = \sum_{k=2}^{\infty} F_k(t)\ \underline{r}^{[k]}(t)$ then for any $k > 0$ we have:

$$d/dt\ \underline{r}^{[k]}(t) = \sum_{j=k+1}^{\infty} R_{kj}\ \underline{r}^{[j]}(t)$$

and thus R_{kj} are implicitely defined.

Theorem 1 (Loparo and Blankenship [18])

Consider the non-linear system $d/dt\ \underline{x}(t) = \underline{f}(\underline{x}(t), t)$ with $\underline{x}_0 \varepsilon R^n$ with $\underline{f}$ as in definition 3, and let $\varepsilon > 0$, $T < \infty$ be given. Then the non-linear system has a unique continuous solution and there exist an integer $p = p(\varepsilon,T,\underline{x}_0)$, a linear map A_p such that the unique solution of the linear differential equation:

$$\frac{d}{dt}\underline{y}_p(t) = A_p(t)\ \underline{y}_p \quad \text{with} \quad \underline{y}_p(0) = \underline{x}_0^{[p]} \quad \text{and :}$$

$$A_p = \begin{bmatrix} F_1(t) & F_2(t) & F_3(t) & \cdots\cdots & F_p(t) \\ \underline{0} & F_{1[2]}(t) & R_{23}(t) & \cdots\cdots & R_{2p}(t) \\ \underline{0} & \underline{0} & F_{1[3]}(t) & \cdots\cdots & R_{3p}(t) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \underline{0} & \underline{0} & \cdots\cdots & \cdots\cdots & F_{1[p]}(t) \end{bmatrix}$$

satisfies the property:

$$\sup_{0 \leqq t \leqq T} \| \underline{x}(t) - \tilde{\underline{x}}_p(t) \| < \varepsilon$$

where $\tilde{\underline{x}}_p(t) = [\ I_{nxn} : \underline{0}\ ]\ \underline{y}_p$

Proof The proof of this theorem can be found in [18] in complete detail.

In essence the theorem guarantees that for a large class of non-linear analytic systems, we can always find a sufficiently close approximation by finding the solution of a high dimensional linear system whose parameters are computable in terms of the coefficients of the Volterra series expansion of the non-linear map. The forcing initial state is related through a tensor power to the original initial state of the non-linear system. The projection of the trajectory solution, for this high dimensional system, into the original state space produces the desired approximation. Loparo and Blankenship [18] utilize this result for estimating the domain of attraction of non-linear feedback systems.

III FEASIBLE REGIONS FOR NON-LINEAR SYSTEMS

In this section we make use of Loparo's result to approximate arbitrarily close the feasible region of a non-linear system whose initial state is bounded by a compact GP. The theorem given in this section is a simple generalization of theorem 1 and follows from the results in section II.

Consider the non-linear system:

$$\frac{d}{dt}\underline{x}(t) = \underline{f}(\underline{x}(t), t) \quad \text{with} \quad \underline{x}_0 \varepsilon \Sigma_0 \tag{3.1}$$

where Σ_0 is a compact GP defined by:

$$\Sigma_0 = \{\underline{x} \varepsilon R^n : < \underline{x}^{(q)}, \underline{h}_{i0} > \leqq 1;\ i=1,2,\dots,M\} \tag{3.2}$$

Then the Σ_0-feasible set at time T can be approximated to within a desired error by utilizing the result of theorem 1 since each and everyone of the feasible trajectories arising from an initial state in Σ_0 can be approximated. The set of all the approximating solutions at time T, constitutes an approximate representation of the feasible set originated by Σ_0

Theorem 2 Consider the non-linear system (3.1) with $\underline{f}$ as in definition 3 and the initial state x_0 confined within the bounds of a compact set described by (3.2). Then the set:

$$\{ \underline{x} \varepsilon R^n : < \underline{x}^{(p)}, \tilde{\underline{h}}_i(T) > \leqq 1 ;\ i= 1,2,\dots, M \} \tag{3.3}$$

where $\tilde{\underline{h}}_i(T)$ is given by the unique solution at time T of the linear differential equation :

$$\frac{d}{dt}\tilde{\underline{h}}_i(t) = - A_p'(t)\, \tilde{\underline{h}}_i(t) \quad \text{with} \quad \tilde{\underline{h}}_i(0) = [\, \underline{h}_{i0};\ \underline{0}\,] \tag{3.4}$$

$p = p(\varepsilon,T,\Sigma_0)$ (with ε a prespecified approximation error) $> q$

$p = q$ if $p(\varepsilon,T,\Sigma_0) \leqq q$

approximates to within a desired ε the Σ_0-feasible set of system (3.1) subject to initial state constraints of type (3.2).

Proof Theorem 2 is a direct consequence of Theorem 1 established by Loparo and Blankenship. It should be noted however that because of the possible difference that may arise among the parameters $p(\varepsilon,T,\Sigma_0)$ and q, one should always choose the greater,so that (3.4) always makes sense and the specified error can even be made smaller. Notice that p can also be considered as a function of each index i. It is not difficult to see that among all possible support vectors defining the initial state set, one should choose as a value of p that which is rendered maximum as a function of the support vectors considered one by one.

Even if the initial state set is not a GP, as often occurs when trascendental functions are used to define such sets, it can be shown under certain mild conditions that these can also be approximated by a GP to within an ε. Obviously in such cases our approach would require a second approximation to the problem of finding a description for the Σ_0-feasible set.

We now present some simple examples which underline some of the salient features of non-linear maps, induced by non-linear systems, on compact, convex sets. It will be seen how convexity is destroied by the action of a non-linear map.

Example 1 Consider the following simple non-dynamic example given by the action of a non-linear map on a convex polyhedron.

Let:

$$y_1 = x_1^2 + x_2$$

$$y_2 = \sqrt{2}\, x_1$$

and the set:

$$\{ \underline{x} \in R^2 : \; < \underline{x} , \underline{h}_i > \leqq 1 \; ; \; i=1,2,3,4 \}$$

with:

$$\underline{h}'_1 = (1,1) \; ; \; \underline{h}'_2 = (1,-1)$$

$$\underline{h}'_3 = (-1, 1) \quad \underline{h}'_4 = (-1,-1)$$

Fig. 1 shows the first and second order approximations to the set into which the polyhedron is mapped by action of the non-linear relation defined above.

The above example clearly shows how convexity of a compact set is eliminated by a non-linear map even in the simplest case of non-linearity. The Carleman linearization approach for this example rapidly renders an image which is practically impossible to differentiate from the actual exact image set.

Example 2 Consider the Van der Pol differential equation:

$$x'' + x - \quad (1 - x^2)\, x' = 0$$

which can also be written in state space form in the usual way :

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = - x_1 + \quad (1 - x_1^2) x_2$$

Suppose we are given a set of initial states represented by the unit circle:

$$x_1^2 + x_2^2 \leqq 1$$

which is a Generalized Polyhedron in R^2 with support vector given by:

$$\underline{h}'_1 = (0,0,1,0,1)$$

The feasible set can then be easily computed using our preceeding approach by succesive approximations, i.e with p= 2,3,.. , for a certain prespecified terminal time T. The first approximation to the reachable set (p=2) turns out to be an ellipse while from the third approximation on, we obtain a figure closely resembling the limit cicle contour. The procedure is easily implemented on a digital computer given the linear nature of the equations. Figure 2 gives an idea of the shape of the feasible set for the Van der Pol system. The non-connectedness of this set is clearly shown in this figure.

The above examples show how convexity and simple connectedness of a compact initial state set are lost by the action of non-linear maps either generated or not by a non-linear differential equation. However, it should be emphasized that this need not be the case in every situation. The followin example indicates that in certain case these two important features of some sets, could be preserved by the action of the non-linear system.

Example 3 The second order equation:

$$y'' + ay + by^3 = 0 \quad ; \quad \text{with} \quad a > 0 \; , \; b > 0$$

can be rewritten as a system :

$$y' = x$$

$$x' = -ay - by^3$$

It is easy to see that the initial state set $x^2 + y^2 \leqq 1$ generates , afeter a sufficiently large time, a feasible set represented by the region:

$$x^2 + ay^2 + \frac{1}{2} by^4 \leqq K$$

which is a convex, connected region closely resembling an ellipse.

IV CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH

This paper has examined an approach for computing the feasible regions of a non-linear analytic system. The approach is based in a Carleman's linearization technique and it only involves computations associated with linear dynamic systems for which a vast amount of computer based resources are readily available. The drawbacks of the method rely on the fact that a high dimensional linear system has to be solved for obtaining the required approximation. On the other hand the computations associated with this large sistem are substantially facilitated due to the sparse nature of the system matrix (tipically a 20x20 matrix generated by the procedure, contains only about 70 non-zero elements).

The results outlined in this paper have implications in several areas related to the study and control of non-linear systems. First of all, the computation of the domain of attraction for non-linear systems([18]) can benefit from the introduction of the idea of starting directly with a generalized polyhedron as an intial guess for this region. This paper has shown that the hansling of such sets is mathematically convenient and requires no substantial extra effort.

The feasible set concept has, by itself, a number of implications in set-estimation and set-control problems. This paper simply widens the range of applicability of the theory by demonstrating that closely enough approximations can be worked for "modified target sets", "attainability sets" etc. (See [6]) related to set-reachability problems associated with non-linear systems.

The method allows for research into the field of target set reachability problems by non-linear feedback systems. Approximate control strategies can be obtained by considering parametric control of the linear approximation system when the feedback map is included explicitely in the original non-linear equations. (See [18] for this simple extension).

The idea of utilizing Generalized polyhdera in some of the existing problems related to set-theoretic estimation and control should be persued further due to the flexibility that these sets offer for the handling of the mathematical equations.

V REFERENCES

[1] Witsenhausen, H., "Set of Possible States of Linear Systems given perturbed Observations," IEEE Transactions on Automatic Control, Vol. AC-13, No. 5 pp 556-558. October 1968.

[2] Podchukaev, V., "Determining the Region of Feasible States for a Non-Stationary Linear System," Automation and Remote Control, Vol. 7 pp 187-189. July 1971.

[3] Sira-Ramirez,H., " On a New Class of Semi-linear Systems and their Applications," Proceedings of the Sixteenth Annual Allerton Conference on Communications, Control and Computing. Allerton House, Monticello, Ill. October 1978.

[4] Schweppe, F., "Recursive State Estimation: Unknown but Bounded Errors and System Inputs," IEEE Transactions on Automatic Control, Vol. AC-13 No. 1, pp. 22-28. February 1968.

[5] Hnyilicza, E., " A Set Theoretic Approach to State Estimation," M.S.

Thesis. Massachusetts Institute of Technology. Cambridge 1967.

[6] Bertsekas, D., and Rhodes,I., " On the Mini-max Reachability of Target Sets and Target Tubes," Automatica, Vol. 7 pp 233-247. Pergamon Press.

[7] Schlaepfer, F., and Schweppe.,F., "Continuous Time State Estimation under Disturbances Bounded by Convex Sets," IEEE Transactions on Automatic Control. Vol. AC-17 No. 2, April 1972.

[8] Sira-Ramirez, H., "Set-Theoretic Control of Large Scale Uncertain Systems," PhD. Thesis. Massachusetts Institute of Technology. Cambridge 1977.

[9] Barmish, B., and Schmittendorf, W., "A Necessary and Sufficient Condition for Local Constrained Controllability of a Linear System," Proceedings of the Sixteenth Annual Allerton Conference on Communications Control and Computing. Allerton House, Monticello, Ill. October 1978

[10] Barmish, B., "A New Notion of Robustness for Perturbed Linear Dynamic Systems," Proceedings of the 7th Triennial World Congress of the International Federation of Automatic Control. Helsinki 1978.

[11] Barmish, B., "Envelope Correspondence and its Applications to the Design of Guaranteed Error Controllers," SIAM J. on Control and Optimization. Vol. 15., pp. 1001-1015, 1977.

[12] Barmish, B., "A Mini-max Theorem for Guaranteed Terminal Error," IEEE Transactions on Automatic Control, Vol. AC-20, No. 6 pp 797-799. December 1975.

[13] Schmittendorf, W. and Barmish, B., " Constrained Controllability," Proceedings of the IEEE Conference on Decision and Control., San Diego, 1978.

[14] Chukwu, E., and Silliman, S., "Complete Controllability to a Closed Target Set," Journal of Optimization Theory and Applications. Vol. 21 pp. 369-383., 1977.

[15] Sira-Ramirez, H., "Evolution of Generalized Uncertainties in Linear Systems," Proceedings of the 1979 Conference on Information Sciences and Systems, The John Hopkins University. Baltimore. MA. March 1973.

[16] Sira-Ramirez, H., " Evolution of Generalized Set-Theoretic Uncertainties in Bi-linear Systems," Proceedings of the 1979 International Symposium on Circuits and Systems. Tokio, Japan. July 1979.

[17] Brockett, R., " On the Reachable Sets for Bi-linear Systems," in Variable Structure Systems with Applications to Economics and Biology. Lecture Notes in Economics and Mathematical Systems, Vol. 111. Springer Verlag. New York. 1975.

[18] Loparo, K., and Blankenship, G., "Estimating the Domain of Attraction of Non-linear Feedback Systems," IEEE Transactions on Automatic Control , Vol. AC-23, No. 4, pp. 602-608. August 1978.

[19] Brockett, R., "Volterra Series and Geometric Control Theory," Automatica. Vol. 12, pp.167-176. Pergamon Press 1976.

[20] Brockett, R., "Lie Algeberas and Lie Groups in Control Theory," in Geometric Methods in Control Thery , (D.Q. Mayne and R.W. Brockett, Eds.) Reidel, Dordrecht, Holland. 1973.

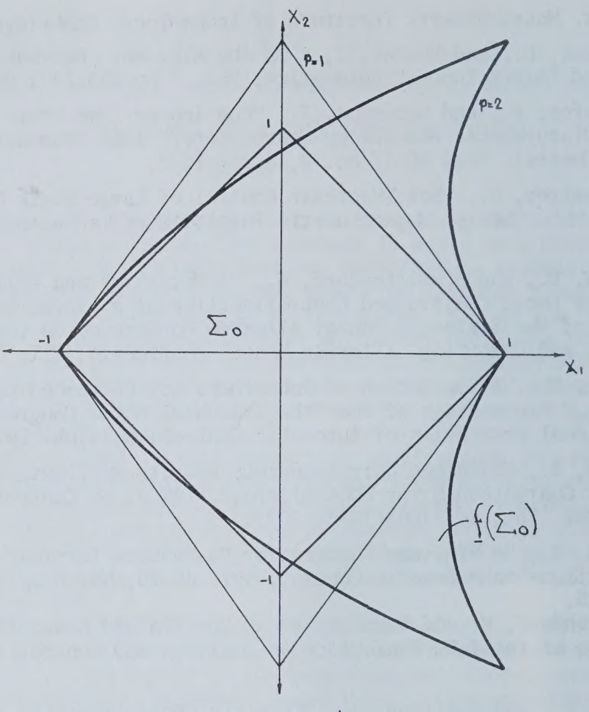

X2
p=1
p=2
1
Σo
-1
1
X1
f(Σo)
-1

FIGURE 1

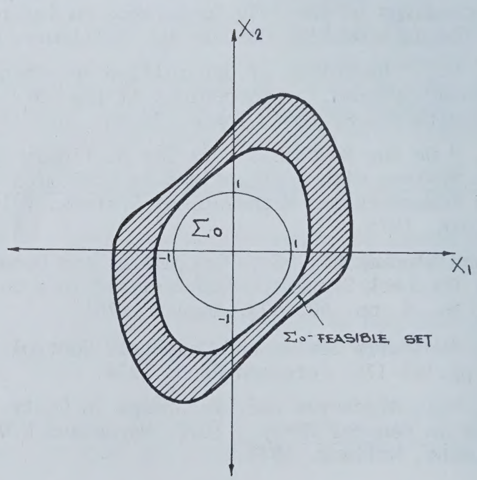

X2
1
Σo
-1
1
X1
-1
Σo-FEASIBLE SET

FIGURE 2
Σo-Feasible Set far van der pol system

IMPROVED ESTIMATORS FOR NONRANDOM WAVEFORMS IN THE PRESENCE OF NOISE*

C. A. POMALAZA, C. D. MCGILLEM
PURDUE UNIVERSITY
WEST LAFAYETTE, INDIANA 47907

ABSTRACT

The use of improved statistical estimators is proposed for the estimation of unknown non-random signals in the presence of an additive noise process. A mini-max approach that makes use of partial information about the signal gives estimates that exceed the performance of more conventional estimators; e.g., M.L. estimate in the case of a gaussian noise process. For a low SNR a considerable reduction of the mean square error is obtained at the expense of introducing a small bias.

INTRODUCTION

There are many communication problems in which, by their very nature, some of the parameters or signals to be estimated are nonrandom. Although usually there exists some partial information about them, this knowledge is not efficiently used by most conventional estimation procedures such as the maximum likelihood method. There are even cases in which the blind application of those methods gives estimates that contradict the partial information about the true nature of the parameters. This situation can become worse when the signal-to-noise ratio (SNR) is small and all of the a priori information should be used in order to improve the performance of the estimators. To cope with these problems several approaches can be taken. One of them is to use the a priori information to set up a probabilistic model for the parameter and then use procedures such as the Bayesian methods to estimate random variables. Another approach is to keep the nonrandom nature

---

*This research supported by AFOSR under Contract No. F 49620-79-C-0084 and N.I.H. Grant No. NS-15347.

of the parameters and use the a priori information to define a space that contains the parameter, then devise an estimation procedure on this space, whose performance can be evaluated in some sense, usually the mean square error. In this paper we will take the later approach. In the signal model that follows the final parameter estimation is reduced to an univariate problem. Some other conditions can lead to a multivariable case whose solution could be more sophisticated, according to the particular characteristics of the problem. However, the main concepts of the univariate model can be extended readily to the multivariable case.

THE MODEL

Assume that an observed signal r(t) is the sum of an unknown non-random signal s(t) and a zero mean random process n(t), i.e.

$$r(t) = s(t) + n(t) \qquad 0 < t < T \tag{1}$$

If n(t) is a gaussian random process then the maximum likelihood estimate (M.L.E.) of s(t) is r(t) [1]. This estimate is unbiased and efficient, thus no other unbiased estimate can have less mean square error (m.s.e.). Now, let it be assumed that the noise can be approximated as a bandlimited stationary random process and the signal s(t) is also approximately bandlimited with a bounded energy content. Also, as in most practical problems, let it be assumed that s(t) and the noise covariance $K_n(\tau)$ are approximately time limited. These assumptions may be summarized as follows.

$$K_n(\tau) \simeq 0 \qquad \text{for } |\tau| > T \tag{2}$$

$$S_n(f) = F_n(K_n(\tau)) \simeq 0 \qquad \text{for } |f| > f_{max} \tag{3}$$

$$s(t) \simeq 0 \qquad \text{for } t < 0 \text{ and } t > T \tag{4}$$

$$S(f) = F_n(s(t)) \simeq 0 \qquad \text{for } |f| > f_{max} \tag{5}$$

$$\int_0^T s^2(t)dt = \int_{-f_{max}}^{f_{max}} |S(f)|^2 df \leq E \tag{6}$$

where F_n stands for the normalized Fourier Transform. With these assumptions it can be shown [1] that the eigenfunctions of $K_n(\tau)$ are closely approximated by the normalized complex exponentials; i.e.,

$$\phi_i(t) = \frac{1}{\sqrt{T}} \exp(\pm j2\pi f_i t) \qquad 0 < t < T \tag{7}$$

$$f_i = \frac{i}{T} \qquad \begin{array}{l} i = 0,1,\ldots,k \\ f_k = f_{max} \end{array}$$

Calling r_i, s_i, n_i the projections of r(t), s(t), and n(t) along $\phi_i(t)$, it follows from the assumptions above that,

$$E\{n_i n_j^*\} = \begin{cases} S_n(f_i) & i=j \\ 0 & i\neq j \end{cases} \tag{8}$$

$$\begin{array}{llll} E\{\text{real}\ (n_i)\ \text{imag}\ (n_j)\} = 0 & \text{for all } i \text{ and } j & \text{(a)} \\ E\{\text{real}\ (n_i)\ \text{real}\ (n_j)\} = 0 & \text{for } i\neq j & \text{(b)} \\ E\{\text{imag}\ (n_i)\ \text{imag}\ (n_j)\} = 0 & \text{for } i\neq j & \text{(c)} \\ E\{[\text{real}\ (n_i)]^2\} = E\{[\text{imag}\ (n_i)]^2\} = S_n(f_i)/2 & & \text{(d)} \end{array} \tag{9}$$

(8) and (9) make it possible to estimate separately the real and imaginary components of s_i. Those estimates are then combined to form a complex function whose inverse Fourier transform gives an estimate $s_k(t)$ of s(t). The problem has now become simplified to the univariate parameter estimation model

$$r = s + n \tag{10}$$

where n is a r.v. with zero mean and variance σ^2, r would be the unbiased estimate of s and in the case of gaussian n it would also be the M.L.E. of s.

ESTIMATION METHOD

There are many strategies available to obtain improved estimators of s [2]. Among them is the mini-max principle which will be discussed here. The mini-max principle chooses as an estimate of s the value $\hat{s}_{mm}$ such that the supremum of the risk function $R(\hat{s},s) = E(\hat{s}-s)^2$ is minimum; i.e.,

$$R(\hat{s}_{mm},s) = \min_{\hat{s}} \sup_{s\in\Omega} R(\hat{s},s) \tag{11}$$

where Ω is the region that contains s and it is defined based on the partial knowledge available about s. For s amplitude limited Ω becomes,

$$\Omega = \{s: s^2 < E^2\} \tag{12}$$

The mini-max solution for the linear case; i.e., $\hat{s}_{mm} = a\, r$, is

$$\hat{s}_{mm} = \frac{r}{1 + \sigma^2/E^2} \tag{13}$$

Defining the normalized m.s.e. as

$$\varepsilon_N^2(\hat{s}) = \frac{\text{m.s.e. } (\hat{s})}{\sigma^2} \tag{14}$$

then

$$\varepsilon_N^2(\hat{s}_{\text{unbiased}}) = 1 \tag{15}$$

For the mini-max solution

$$\varepsilon_N^2(\hat{s}_{mm}) = \frac{k^4 + \mu^2}{(k^2 + 1)^2} \tag{16}$$

where $k = \frac{E}{\sigma}$ and $\mu = \frac{s}{\sigma}$.
Defining the normalized bias as

$$b_N(\hat{s}) = \frac{E(\hat{s}) - s}{s}, \quad s \neq 0 \tag{17}$$

then

$$b_N(\hat{s}_{\text{unbiased}}) = 0 \tag{18}$$

and

$$b_N(\hat{s}_{mm}) = \frac{-1}{k^2 + 1} \tag{19}$$

Figures 1 and 2 illustrate $\epsilon_N^2(\hat{s}_{mm})$ and $b_N(\hat{s}_{mm})$ for different values of C, where $k = C\mu$, $C \geq 1$. The case C = 1; i.e., $E^2 = s^2$ gives the "best" improved estimate of s, however, this estimate is impractical because it assumes a knowledge of s. Figure 1 shows the reduction of the m.s.e. for signal-to-noise ratios around 0 dB while the amount of bias is still small. This mini-max approach is a linear procedure. Nonlinear methods can also be used. One of them, suggested by Thompson [4] performs reasonably well for small SNRs, and leads to the estimate

$$\hat{s}_T = \frac{r^3}{r^2 + \sigma^2} \tag{20}$$

For the gaussian noise case this estimate shows a reduction in the m.s.e. for small SNRs with the additional advantage that the amount of bias stays small for a broader range than the mini-max solution. This is illustrated in Figures 3 and 4.

APPLICATION

Figures 5-7 show a practical application of the mini-max method to the estimation of visual evoked responses of the brain to a light estimulus. Figure 5 shows the spectrum $|S(f)|^2$ of the evoked response and the power spectral density N(f) of the ongoing EEG (considered here as noise). Figures 6 and 7 show some actual estimates and the errors involved.

CONCLUSION

A priori information about nonrandom parameters can be used in estimation procedures like the mini-max method, in order to obtain results that are better (in the m.s.e. sense) than those obtained by more conventional procedures. Information such as bounds on amplitude, ranges of signal-to-noise ratios or regions where the parameters can be located, can be used to obtain improved estimates although some desirable properties such as unbiasedness may be lost.

REFERENCES

[1] Van Trees, H. L., Detection, Estimation, and Modulation Theory, Part 1, Wiley 1968.

[2] Bibby, John and Helge Toutenburg, Prediction and improved estimation in linear models, Wiley, 1977.

[3] Lauter, H., "A mini-max linear estimator for linear parameters under restrictions in form of inequalities," Mathematische Operationsforschung und statistik, 1975, pp. 689-696.

[4] Thompson, J. R., "Some shrinkage techniques for estimating the mean," Journal of the American Statistical Association, March 1968, pp. 113-122.

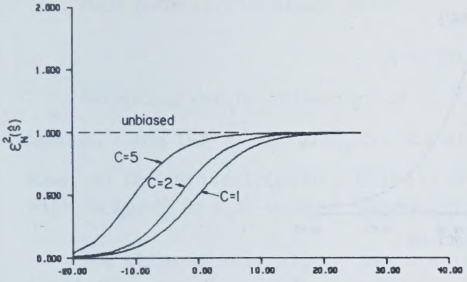


Fig. 1. Normalized mean square error of the unbiased and three mini-max estimators.

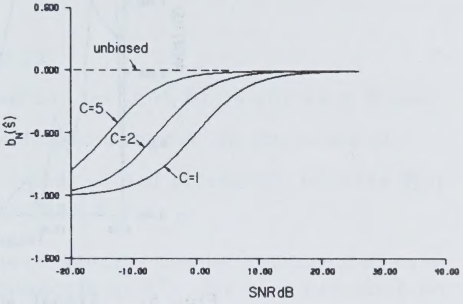


Fig. 2. Normalized bias of the unbiased and three mini-max estimators.

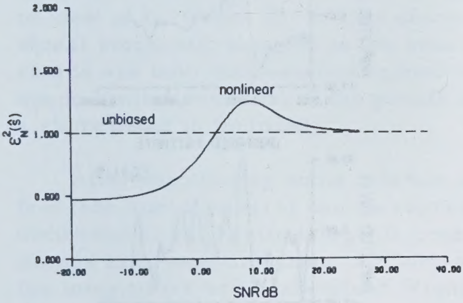


Fig. 3. Normalized error of the unbiased and a nonlinear estimator (see text).

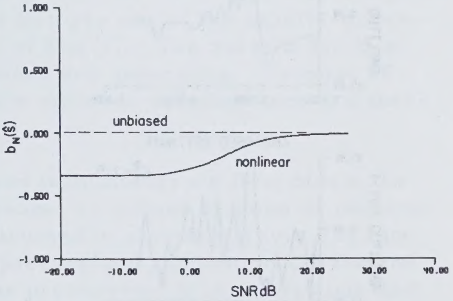


Fig. 4. Normalized bias of the unbiased and a nonlinear estimator.

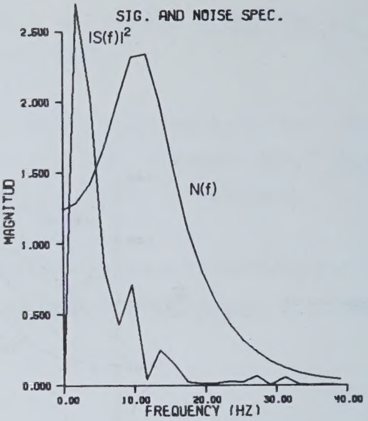


Fig. 5. Signal and noise spectra
SNR = -5 db

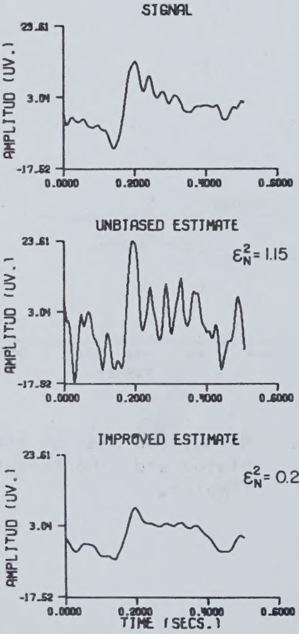


Fig. 6. Visual evoked response and a sample of its unbiased and mini-max estimators (C=1).

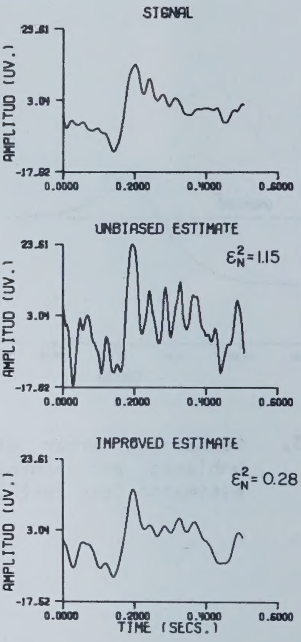


Fig. 7. Visual evoked response and a sample of its unbiased and mini-max estimators (C=2).

APPROXIMATION OF HILBERT-SPACE-VALUED STOCHASTIC INTEGRALS BY FINITE-DIMENSIONAL ONES

NADAV BERMAN
WILLIAM L. ROOT
The University of Michigan
Ann Arbor, Michigan 48109

ABSTRACT

Adopting the terminology of L. Gross, let (I, H, B) be abstract Wiener space, and let $\int_0^t \Phi(s)dW_s$ be the stochastic integral, in the sense of Kuo, of the nonanticipating Hilbert-Schmidt valued stochastic process $\Phi(s)$ with respect to a B-valued Wiener process $\{W_t\}_{t \geq 0}$.

We show in this paper that the above integral can be represented as a limit of a sequence of Ito stochastic integrals in R^n. We also establish an alternative definition of this integral which proves to be useful in some approximation problems.

INTRODUCTION

The results presented in this paper are a part of some research on approximation of stochastic differential equations in Hilbert space in connection with filtering and identification problems. We adopt here the point of view of L. Gross [2] and we choose to make use of the infinite-dimensional stochastic integral in the sense of Kuo [7]. The reasons for this choice are both mathematical convenience and generality. Because of space limitations most of the proofs are omitted, although here and there a short proof is included.

After introducing some notation and terminology we first obtain the fact that Kuo's integral can be represented as a limit of sums of infinite-dimensional stochastic integrals constructed in a certain way. The integrands are nonanticipating, Hilbert-space-valued random processes and the integrators are real-valued Wiener processes. It is pointed out that this limit may serve as an alternative definition of a stochastic integral. This is followed with the result that this limit can be approximated by a sequence of finite-dimensional Ito integrals. These results are all for the case that the integrand satisfies the usual square-integrability condition. Similar results hold for a (standard) weaker integrability condition on the integrand.

PRELIMINARIES

Abstract Wiener Space and Banach-Space-Valued Wiener Process

Definition 1. Let H be a real separable Hilbert space with the inner product $(\cdot,\cdot)_H$ and norm $||\cdot||_H$. For each t > 0, μ_t is defined to be a Gaussian cylinder set measure in H if

$$\mu_t(C) = \left(\frac{1}{2\pi t}\right)^{\frac{n}{2}} \int_{\mathcal{B}_n} \exp - \left(\frac{||x||_H^2}{2t}\right) m(dx),$$

where C is a cylinder set in H, that is

$$C = \{x \in H: ((x,e_1)_H, \ldots, (x,e_n)_H) \in \mathcal{B}_n\}, \ \{e_i\}_{i=1}^n$$

is some orthonormal set in H, $\mathcal{B}_n$ is a Borel set in R^n, and m is the Lebesgue measure in R^n. It is well known that μ_t is a finitely additive set function on the algebra generated by cylinder sets in H but is not countably additive there and hence cannot be extended to a probability measure on the Borel sets in H. However μ_t induces a countably additive cylinder set measure on the algebra generated by cylinder sets in a certain Banach space B containing H as a subset. We discuss this shortly below.

Definition 2. Let $||\cdot||_B$ be a seminorm on H and let $\mathcal{F}$ be the collection of all finite-dimensional orthogonal projections in H. $||\cdot||_B$ is said to be a measurable seminorm iff for each $\epsilon > 0$ there exists $Q_o \in \mathcal{F}$ such that $\mu_t\{||Qx||_B > \epsilon\} < \epsilon$ for all $Q \in \mathcal{F}$ such that $Q \perp Q_o$.

Assume $||\cdot||_B$ is a measurable norm on H, denote by B the completion of H with respect to $||\cdot||_B$ and let I be the natural injection of H into B. Let $I^*: B^* \to H^* = H$ be the adjoint of I. By means of the injections I, I^* we may imbed H in B and B^* in $H^* = H$. Thus we have $B^* \subset H^* = H \subset B$. Note that B^* is dense in H^*, i.e., $I^*(B^*)$ is dense in H^*. The triple (I, H, B) is called an abstract Wiener space (AWS). The cylinder set measure $\bar{\mu}_t$ in B is defined as follows: let $\{y_i\}_{i=1}^n$ be an orthonormal set in $R(I^*)$ (range of I^*), $\mathcal{B}_n$ - a Borel set in R^n and let $C = \{x \in B: ((I^{*-1}y_1)(x), \ldots, (I^{*-1}y_n)(x)) \in \mathcal{B}_n\}$ then $\bar{\mu}_t(C) \triangleq \mu_t\{x \in H: ((x, y_1)_H, \ldots, (x, y_n)_H) \in \mathcal{B}_n\}$. It is known that $\bar{\mu}_t$ is countably additive on the algebra generated by the cylinder sets in B, hence it extends uniquely to a probability measure on the σ-algebra generated by the cylinder sets in B (= Borel sets in B). We denote this measure by p_t. It is easy to see that for $t > 0$ each $y^* \in B^*$ is Gaussian distributed with zero mean and variance $t||I^* y^*||_H^2$. Thus for all $t > 0$ p_t is a Gaussian measure on B.

Let Ω be the collection of all continuous functions ω on $[0, \infty)$ with values in B and $\omega(0) = 0$. Then it can be shown that there exists a unique probability measure P on the σ-algebra $\mathcal{A}$ generated by the coordinate functions W_t ($W_t(\omega) = \omega(t)$, $t \geq 0$) such that the process W_t has independent increments and if $t > s$ then $W_t - W_s$ is distributed according to p_{t-s}.

W_t is called a Wiener process in B. We note that W_t is a Markov process starting at the origin with transition probabilities $p_{t,s}(x, A) \overset{\Delta}{=} p_{t-s}(A-x)$, $t > s$, $x \in B$, $A \in \mathcal{B}(B)$ and with the initial probability $p_o(A) = \chi_A(0)$.

The Normal Weak Distribution

For general definition and discussion of weak distribution, see Gross [2]. Here we only need the following special class of weak distribution.

Definition 3. Let $\tilde{n}_t$ be a weak distribution on a real separable Hilbert space H. Then $\tilde{n}_t$ is said to be a normal weak distribution in H with parameter $t > 0$ if for all $h \in H$ $(n_t h)$ is a Gaussian random variable on $(\Omega, \mathcal{A}, P)$ (the probability space associated with the representative n_t of $\tilde{n}_t$) with variance $t\|h\|_H^2$ and mean zero. A particular representative of $\tilde{n}_t$ is introduced in the following lemma.

Lemma 1 (Gross [2]). Let (I, H, B) be AWS and let $\tilde{n}_t$ be the normal weak distribution on H with parameter $t > 0$. Then there exists a representative n_t of $\tilde{n}_t$ such that $n_t: H \to L_2(B, \mathcal{B}(B), p_t)$. Moreover, if $h \in R(I^*)$, then $(n_t h)(x) = (I^{*-1}h)(x)$, $x \in B$.

In the next theorem we need the following terminology: Let f be a complex valued function on H. We say that f is a cylinder set function on H if $f(x) = \phi((x, e_1)_H, \ldots, (x, e_n)_H)$ where ϕ is a complex valued function on R^n and $\{e_i\}_{i=1}^n$ is some orthonormal set in H. If ϕ is measurable we associate with f a random variable $\tilde{f}$ on $(B, \mathcal{B}(B), p_t)$ defined by $\tilde{f}(x) = \phi((n_t e_i)(x), \ldots (n_t e_n)(x))$. If f is measurable and $Q \in \mathcal{F}$ then $f_Q \overset{\Delta}{=} f \circ Q$ is a cylinder set function and we denote by $\tilde{f}$ the limit in p_t (if it exists) of $\{\tilde{f}_Q\}_{Q \in \mathcal{F}}$ as $Q \overset{s}{\to} I_H$ through $\mathcal{F}$.

Theorem 1 (Gross [4]). Let f be a continuous complex valued function on B and let g be the restriction of f to H then $\tilde{g}$ exists and $\tilde{g} = f$ p_t — a.e.

Stochastic Integration

As was mentioned before we consider here stochastic integration in the sense of Kuo. Let (I, H, B) be AWS. Then in the rest of this section it is assumed that B has the following property: there exists a sequence $\{Q_n\}_n$ of finite dimensional orthogonal projections on H such that $Q_n \overset{s}{\to} I_H$ and such that for each n Q_n extends to a projection $\underset{\sim}{\tilde{Q}}_n$ on B, $\|\tilde{Q}_n\|_{BB} \leq 1$ and $\|\tilde{Q}_n x - x\|_B \to 0$ for all $x \in B$. For the general case see Kuo [7]. We take $(\Omega, \mathcal{A}, P)$, W_t to be as defined above.

Let $\Phi(s, \omega) : R_+ \times \Omega \to L_{(2)}(H)$ be a nonanticipating stochastic process (we often write $\Phi(s)$), adapted to the increasing family $\{\mathcal{A}_t\}_{t \geq 0}$

of sub σ-algebras of $\mathcal{Q}$ generated by W_t, i.e., $\mathcal{Q}_t = \{\sigma\text{-algebra generated by } W_s,\ 0 \leq s \leq t\}$. We write $\Phi(s) \in \mathcal{L}(L_{(2)}(H))$. $L_{(2)}(H)$ is the vector space of the Hilbert-Schmidt operators on H with the Hilbert-Schmidt norm $||\circ||_2$. Assume

$$E \int_0^{t_1} ||\Phi(s)||_2^2 ds < \infty$$

for a fixed $t_1 \in R_+$. Then one defines a stochastic integral

$$\int_0^t \Phi(s)dW_s, \quad t \in [0, t_1],$$

in the sense of Kuo (similarly to the Ito integral) which has the following properties:

(a) $x(t) = \int_0^t \Phi(s)dW_s$ is a continuous H-valued martingale and hence for all $\epsilon > 0$, $s \leq t_1$

$$P\{ \sup_{0 \leq t \leq s} || \int_0^t \Phi(\sigma)dW_\sigma || > \epsilon\} \leq \frac{1}{\epsilon^2} E|| \int_0^s \Phi(\sigma)dW_\sigma ||_H^2$$

(b) For all $\epsilon > 0, N > 0$, $P\{ || \int_0^t \Phi(s)dW_s ||_H > \epsilon\}$

$$\leq P\{ \int_0^t ||\Phi(s)||_2^2 ds > N\} + \frac{N}{\epsilon^2}$$

(c) $E \int_0^t \Phi(s)dW_s = 0$, $E|| \int_0^t \Phi(s)dW_s ||_H^2 = E \int_0^t ||\Phi(s)||_2^2 ds.$

Remark. It is shown as in the one dimensional case that this stochastic integral may be extended so that it includes integrands which require only that

$$\int_0^t ||\Phi(s)||_2^2 ds < \infty \qquad P - \text{a.e.}$$

APPROXIMATION OF THE STOCHASTIC INTEGRAL

Using ideas similar to those in Kuelbs [6] we show that the B-valued Wiener process $\{W_t\}_{t \geq 0}$ admits a series representation. Thus we have the following theorem.

Theorem 2. Let (I, H, B) be AWS, $\{W_t\}_{t \geq 0}$ be a B-valued Wiener process defined on $(\Omega, \mathcal{Q}, P)$. Let $\{e_i\}_{i=1}^{\infty}$ be any orthonormal basis for H, then there exists a sequence $\{W_t^i\}_{i=1}^{\infty}$, $t \geq 0$ of real-valued Wiener processes defined on $(\Omega, \mathcal{Q}, P)$ such that (a) For each t, W_t^i's are independent,

(b) $E(W_t^i) = 0, \quad E|W_t^i|^2 = t$ for all t, $i = 1, 2, \ldots,$

(c) $\lim_{n \to \infty} \sum_{i=1}^n W_t^i e_i$ exists in the B-topology P — a.e. and

$$W_t(\omega) = \sum_{i=1}^{\infty} W_t^i(\omega) e_i \qquad P - \text{a.e.}$$

Proof. The proof is a consequence of Theorem 1 and Ito-Nisio theorem on convergence of sums of B-valued independent random variables (see Ito-Nisio [5]).

Remark. In the above series, for each i $W_t^i(\omega)$ is taken to be $(n_t e_i)(W_t(\omega))$ $\omega \in \Omega$ and n_t is the representative of the normal weak distribution as has been introduced above. We remark here that we take $n_t e_i$ to be any representative of the element $n_t e_i \in L_2(B, \mathcal{B}(B), P_t)$.

We now show that the integral in the sense of Kuo can be approximated by finite dimensional Ito integrals. As a first step we need the definition of a stochastic integral where the integrand is an H-valued function and the integrator is a real valued Wiener process.

Using the same definition as in Curtain [1] we have the following in the AWS set up. Let (I HB) be AWS, $\{W_t\}_{t \geq 0}$ be a Wiener process defined on the probability space $(\Omega, \mathcal{A}, P)$ as above. Let W_t^i be as in Theorem 2. Let $\Phi(s) \in \mathcal{L}(L_{(2)}(H))$, $E \int_0^{t_1} ||\Phi(s)||_2^2 \, ds < \infty$, and $\{e_i\}_{i=1}^{\infty}$ be any orthonormal basis for H. Then as in Curtain [1] $\int_0^t \Phi(s) e_i dW_t^i$ has the following properties

(a) $$E\left(\int_0^t \Phi(s) e_i dW_s^i, \int_0^t \Phi(s) e_j dW_s^j\right)_H = \begin{cases} 0, & i \neq j \\ E \int_0^t ||\Phi(s) e_i||_H^2 ds & i = j \end{cases}$$

(b) $$E\left(\int_0^t \Phi(s) e_i dW_s^i\right) = 0$$

Remarks. (1) This integral extends in the obvious way to include functions $\Phi(s) \in \mathcal{L}(L_{(2)}(H))$ such that $\int_0^{t_1} ||\Phi(s)||_2^2 ds < \infty$ $\quad P - \text{a.e.}$
(2) In the definition of the integral one can take $\Phi(s) \in \mathcal{L}(L(B, H))$, where $L(B, H)$ is the usual Banach space of bounded linear transformations from B into H, such that $E \int_0^t ||\Phi(s)||_{BH}^2 ds < \infty$. Then by Kuo [7] the restriction $\tilde{\Phi}(s)$ of $\Phi(s)$ to H is in $\mathcal{L}(L_{(2)}(H))$ and $E \int_0^{t_1} ||\tilde{\Phi}(s)||_2^2 ds < \infty$. Hence $L(B, H)$ can be viewed as a subset of $L_{(2)}(H)$. Moreover by the property of B, assumed in the previous section, $L(B, H)$ is dense in $L_{(2)}(H)$.

Lemma 2. Let (I, H, B) be AWS and assume $\{e_i\}_{i=1}^{\infty}$ is some orthonormal basis for H. Let $\Phi(s) \in \mathcal{L}(L_{(2)}(H))$ and $E \int_0^{t_1} ||\Phi(s)||_2^2 ds < 0$ for some $t_1 > 0$. Define

$$S_n(t) \triangleq \sum_{j=1}^{n} \int_0^t \Phi(s) e_j dW_t^j, \qquad t \in [0, t_1].$$

Then there exists $S(t) \in L_2(\Omega, \mathcal{A}, P, H)$ such that $S_n(t) \to S(t)$ in $L_2(\Omega, \mathcal{A}, P, H)$ uniformly on $[0, t_1]$. Moreover

$$E\|S(t)\|_H^2 = E \int_0^t \|\Phi(s)\|_2^2 ds \text{ for all } t \in [0, t_1].$$

Remark. If $\Phi(s) \in \mathcal{L}(L(B, H))$ and $E \int_0^{t_1} \|\Phi(s)\|_{BH}^2 ds < \infty$ then

$$E\|S(t)\|_H^2 = E \int_0^t \|\tilde{\Phi}(s)\|_2^2 \le \left(\int_B \|x\|_B^2 \, p_1(dx)\right) \times$$

$$\times E \int_0^t \|\Phi(s)\|_{BH}^2 ds < \infty \quad \text{(see Kuo [7])}.$$

Lemma 3. Let $\Phi(s) \in \mathcal{L}(L(B, H))$ and $E \int_0^{t_1} \|\Phi(s)\|_{BH}^2 ds < \infty$. Assume $\Phi(s)$ is a simple function, and let $\int_0^t \Phi(s) dW_s$ be the integral in the sense of Kuo [7]. Let $S_n(t)$ be as in Lemma 2, then $S_n(t) \to \int_0^t \Phi(s) dW_s$ in $L_2(\Omega, \mathcal{A}, P, H)$ uniformly in $[0, t_1]$.

Proof. Let $s < t$, then by Theorem 2

$$\|\Phi(s)(W_t - W_s) - \sum_{j=1}^{n} (W_t^j - W_s^j) \Phi(s) e_j\|_H \to 0 \qquad P - \text{a.e.}$$

This together with Lemma 2 implies $S_n(t) \to \int_0^t \Phi(s) dW_s$ in $L_2(\Omega, \mathcal{A}, P, H)$ uniformly on $[0, t_1]$.

Lemma 4. Let $\Phi(s) \in \mathcal{L}(L(B, H))$, $E \int_0^{t_1} \|\Phi(s)\|_{BH}^2 ds < \infty$ and let $S_n(t)$ be as in Lemma 2. Suppose $\int_0^t \Phi(s) dW_s$ is the integral in the sense of Kuo, then $S_n(t) \to \int_0^t \Phi(s) dW_s$ in $L_2(\Omega, \mathcal{A}, P, H)$ uniformly on $[0, t_1]$.

Proof. Let $\{\Phi_n(s)\}_n$ be a sequence of simple functions in $\mathcal{L}(L(B, H))$ such that $E \int_0^{t_1} \|\Phi_n(s) - \Phi(s)\|_{BH}^2 ds < \infty$. We have then

$$(E\|\int_0^t \Phi(s) dW_s - S_m(t)\|_H^2)^{\frac{1}{2}} = \|\int_0^t \Phi(s) dW_s - S_m(t)\|_{L_2} = \|\int_0^t \Phi(s) dW_s$$

$$- \sum_{i=1}^{m} \int_0^t \Phi(s) e_i dW_s^i\|_{L_2} \le \|\int_0^t \Phi(s) dW_s - \int_0^t \Phi_n(s) dW_s\|_{L_2} +$$

$$+ \|\int_0^t \Phi_n(s) dW_s - \sum_{i=1}^{m} \int_0^t \Phi_n(s) e_i dW_s^i\|_{L_2} +$$

$$+\|\sum_{i=1}^{m}\int_0^t \Phi_n(s)e_i\,dW_s^i - \sum_{i=1}^{m}\int_0^t \Phi(s)e_i\,dW_s^i\|_{L_2}.$$

For the first term on the right hand side we have

$$E\|\int_0^t \Phi(s)dW_s - \int_0^t \Phi_n(s)dW_s\|^2 = E\int_0^t \|\tilde{\Phi}(s) - \tilde{\Phi}_n(s)\|_2^2\,ds \le$$

$$\le (\int_B \|x\|_B^2 p_1(dx))\; E\int_0^t \|\Phi(s) - \Phi_n(s)\|_{BH}^2\,ds \to 0$$

uniformly on $[0, t_1]$ as $n \to \infty$. The second term approaches zero by Lemma 3 uniformly on $[0, t_1]$ (independently of n). The third term approaches zero as $n \to \infty$ uniformly in m. In fact we have

$$E\,\|\sum_{i=1}^{m}\int_0^t \Phi_n(s)e_i\,dW_s^i - \sum_{i=1}^{m}\Phi(s)e_i\,dW_s^i\|_H^2 = \sum_{i=1}^{m}E\|\int_0^t (\Phi(s)-\Phi_n(s))e_i\,dW_s^i\|_H^2$$

$$= \sum_{i=1}^{m}\int_0^t E\|(\Phi(s)-\Phi_n(s))e_i\|_H^2 ds = \int_0^t E(\sum_{i=1}^{m}\|(\Phi(s)-\Phi_n(s))e_i\|_H^2 ds$$

$$\le \int_0^t E(\sum_{i=1}^{\infty}\|(\Phi(s)-\Phi_n(s))e_i\|_H^2)ds = \int_0^t E\|\tilde{\Phi}(s)-\tilde{\Phi}_n(s)\|_2^2 ds$$

$$\le (\int_B \|x\|_B^2\, p_1(dx) \int_0^t E\|\Phi(s)-\Phi_n(s)\|_{BH}^2 \to 0 \text{ as } n \to \infty$$

uniformly in m and uniformly on $[0, t_1]$.

The following lemma extends Lemma 4 to all of $\mathcal{L}(L_{(2)}(H))$ such that $E\int_0^{t_1}\|\Phi(s)\|_2^2 ds < \infty$.

Lemma 5. Let $\Phi(s) \in \mathcal{L}(L_{(2)}(H))$ and $E\int_0^{t_1}\|\Phi(s)\|_2^2 ds < \infty$. Let $\{S_n(t)\}$ be as in Lemma 2, then $S_n(t) \to \int_0^t \Phi(s)dW_s$ in $L_2(\Omega, \mathcal{A}, P, H)$ uniformly on $[0, t_1]$.

Proof. For a fixed $t \in [0, t_1]$ denoted by M the set of all $\Phi(s) \in \mathcal{L}(L_{(2)}(H))$ such that $E\int_0^t \|\Phi(s)\|_2^2 ds < \infty$, then M is a closed subspace of $L_2(\Omega \times [0, t], P \times m, L_{(2)}(H))$ where m is the Lebesgue measure on $[0, t]$. By Lemma 2 the transformation $\Phi(s) \to S(t)$ is an isometry from M into $L_2(\Omega, \mathcal{A}, P, H)$. The Kuo integral $\int_0^t \Phi(s)dW_s$ is also an isometry between these two spaces, and we proved they agree on the set $\mathcal{L}(L(B, H))$ with $E\int_0^t \|\Phi(s)\|_{BH}^2 ds < \infty$, which is dense in $\mathcal{L}(L_{(2)}(H))$ with $E\int_0^t \|\Phi(s)\|_2^2 ds < \infty$; hence they agree on M. Since $S_n(t) \to S(t)$ uniformly on $[0, t_1]$

we have that $S_n(t) \to \int_0^t \Phi(s)dW_s$ uniformly on $[0, t_1]$ for $\Phi(s) \in \mathcal{L}(L_{(2)}(H))$ and $E\int_0^{t_1} ||\Phi(s)||_2^2 ds < \infty$.

Remark. Lemma 2 may serve as a definition for a stochastic integral. We note that $S_n(t)$ in Lemma 2 converges in $L_2(\Omega, \mathcal{A}, P, H)$ regardless of whether B has the property assumed in the section on stochastic integration or has not. This integral has the properties given above for Kuo's integral.

Using the last lemma we can prove that Kuo's integral can be approximated by a sequence of finite dimensional Ito integrals. We have then

Lemma 6. Let $\Phi(s) \in \mathcal{L}(L_{(2)}(H))$ be such that $E\int_0^t ||\Phi(s)||_2^2 ds < \infty$, and let $\{W_t^i\}_{i=1}^{\infty}$, $t \geq 0$ be as defined before. Put $\bar{W}_t^n = \sum_{i=1}^n W_t^i e_i$ and define

$$\int_0^t Q_n\Phi(s)Q_n d\bar{W}_s^n \triangleq \sum_{j=1}^n \Big(\sum_{i=1}^n \int_0^t (\Phi e_i, e_j)_H dW_s^i \Big) e_j$$

where Q_n is the orthogonal projection on the subspace H_n of H generated by $\{e_i\}_{i=1}^n$ and $\int_0^t (\Phi(s)e_i, e_j)_H dW_s^i$ is the one dimensional Ito integral. Then

$$\int_0^t Q_n\Phi(s)Q_n d\bar{W}_s^n \to \int_0^t \Phi(s)dW_s \text{ in } L_2(\Omega, \mathcal{A}, P, H) \text{ uniformly on } [0, t_1].$$

Remark. Obviously $Q_n\Phi(s)Q_n \in \mathcal{L}(L_{(2)}(H))$ and $E\int_0^{t_1} ||Q_n\Phi(s)Q_n||_2^2 ds < \infty$. Thus it is easy to see that $\int_0^t Q_n\Phi(s)Q_n dW_s = \int_0^t Q_n\Phi(s)Q_n d\bar{W}_s^n$ P — a.e.

We now summarize similar results for the case where the stochastic integral is defined for stochastic processes $\Phi(s)$ in $\mathcal{L}(L_{(2)}(H))$ such that $\int_0^{t_1} ||\Phi(s)||_2^2 ds < \infty$ P — a.e. For this case it is shown that

$$\lim_{n \to \infty} \sum_{i=1}^n \int_0^t \Phi(s)e_i dW_s^i$$ exists in P-measure and that for each $\epsilon < 0$

$$P\{ \sup_{t \in [0, t_1]} || \int_0^t \Phi(s)dW_s - \sum_{i=1}^n \int_0^t \Phi(s)e_i dW_s^i ||_H > \epsilon \} \to 0 \text{ as } n \to \infty.$$

One can then use the above sum as a definition for the stochastic integral. As in the previous case one can show that $\sum_{i=1}^n \int_0^t \Phi(s)e_i dW_s$ converges in probability for arbitrary given AWS and the limit is shown to have continuous sample paths. Using this we have the following lemma.

Lemma 7. Suppose $\Phi(s) \in \mathcal{L}_2(L_{(2)}(H))$, $\int_0^{t_1} ||\Phi(s)||_2^2 ds < \infty$

P — a.e. and let $\bar{W}^n_t$, $n=1, 2, \ldots$, $\{Q_n\}_{n=1}^{\infty}$ be as in Lemma 6. Define

$$\int_0^t Q_n \Phi(s) Q_n d\bar{W}^n_s \triangleq \sum_{j=1}^{n} \left(\sum_{i=1}^{n} \int_0^t (\Phi e_i, e_j)_H dW^i_s \right) e_j, \text{ where}$$

$\int_0^t (\Phi(s)e_i, e_j) dW^i_s$ is the Ito integral in the "weaker" sense. Then for all $\epsilon > 0$ we have

$$P\{ \sup_{t \in [0, t_1]} \|\int_0^t \Phi(s) dW_s - \int_0^t Q_n \Phi(s) Q_n d\bar{W}^n_s \| > \epsilon\} \to 0 \text{ as } n \to \infty.$$

REFERENCES

1. R. F. Curtain, A. J. Pritchard, Infinite Dimensional Linear Systems Theory, Lecture Notes in Control and Information Sciences, Springer Verlag (1978).
2. L. Gross, "Measurable Functions on Hilbert Space," Trans. AMS, v105, pp. 372-390 (1962).
3. L. Gross, "Potential Theory on Hilbert Space," J. Func. Anal., v1, pp. 123-181 (1967).
4. L. Gross, "Abstract Wiener Space," Proc. Fifth Berkeley Sym. Math. Stat. and Prob. v2, pp. 31-42 (1965).
5. K. Ito, M. Nisio, "On the Convergence of Independent Banach Space Valued Random Variables," Osaka J. Math., v5, pp. 35-48 (1968).
6. J. Kuelbs, "Expansions of Vectors in Banach Space Related to Gaussian Measures," Proc. AMS, v27, pp. 364-370 (1971).
7. H.-H. Kuo, Gaussian measures in Banach spaces, Lecture notes in mathematics 463, Springer Verlag (1975).

APPROXIMATE REPRESENTATION OF BANDLIMITED SIGNALS BY ELEMENTARY FUNCTIONS*

A. SONNENSCHEIN and B. DICKINSON
Department of Electrical Engineering and Computer Science
Princeton University
Princeton, New Jersey 08544

ABSTRACT

A novel scheme for the approximation of a bandlimited signal by a weighted sum of elementary functions is presented. The weighting coefficients are derived from the prolate spheroidal wave function expansion coefficients and, consequently, only knowledge of the signal over a finite time interval is needed.

I. INTRODUCTION

There are numerous approaches to generating representations for bandlimited signals [17]. Among them are: (A) the Taylor series expansion, (B) the sampling expansion, and (C) Fourier-type expansions in terms of a set of functions complete and orthonormal in the bandlimited signal space. In our comparison of these approaches the relevant features we consider are the simplicity and rate of decay of the representing functions, the ease of generating the coefficients, the necessary record length, and the sensitivity to noise.

Approach (A) has the advantages of needing only a finite record of the signal and using simple functions (i. e. powers of t) in the representation. It is usually discounted, however, because the expansion is extremely sensitive to errors at the sampling point. Also, the sensitivity to truncation of the series makes the latter a good approximation to the signal only near the sampling point.

Approach (B) involves a sum of translates of the function $\frac{\sin t}{t}$ weighted by samples of the signal [18]. The sampling expansion involves easy but noise-sensitive calculation of the coefficients (merely uniformly spaced samples of the signal). Fortunately, small errors in the coefficients generate only a small error in the final representation [23]. However, the slow $[O(\frac{1}{t})]$ rate of decay of the sinc function requires a large number of terms in the series, which in turn requires knowledge of the signal over a long period of time.

The spherical Bessel function representation [3] and the prolate spheroidal wave function representation [4] are two examples of Approach (C).

The former representation method needs to know the signal waveform over all time to compute the coefficients. The computation involves an integral, which results in some immunity to low-level noise. Although not pointed out in [3], the spherical Bessel functions can be written as finite sums of elementary functions [cf. section IV] having a decay rate of $O(t^{-N})$, so that the spherical Bessel function representation has the form of a sum of functions which decay as $O(t^{-N})$.

The latter representation method makes use of the double orthogonality of the prolate functions to implicitly extrapolate the signal from integrals taken over a finite time interval. The prolate functions are needed to compute the coefficients and to form the representation. The functions decay very quickly [4], but they are very difficult to generate. Both the prolate function representation method and the Bessel function representation method limit their respective L_2 errors. For bandlimited signals, these limits imply limits on the respec-

---

*This work was supported in part by the National Science Foundation through grant ENG77-28523.

tive uniform errors.

The problem we consider in this paper is that of finding a representation for bandlimited signals which has three desireable properties: (1) it utilizes functions that are easy to generate and have quick decay, (2) it computes the coefficients by means of an integral over a finite record, and (3) it bounds the L_2 error due to truncation. We allow for some difficulty in the computation of the integral, as well as the possibility of imperfect reconstruction; that is, the possibility that the energy of the error can be made as small as desired, but that, in certain cases, it cannot be zero. Such a problem formulation is appropriate if, for instance, we have a central station with considerable computational ability attempting to transmit over a medium of given bandwidth to a number of local stations with limited computational ability. Another possible application is the storage of bandlimited waveforms for easy future retrieval.

It is easy to see that none of the methods we have reviewed satisfies the above requirements. The scheme we will subsequently derive does meet these requirements, and it does so by combining the desirable features of Approaches (A) and ~~(B).~~ (C).

Considerations involved in some of our previous work [5] led us to an approximate representation of bandlimited signals which bears a resemblance to Bessel function representation. The coefficients of the expansion we obtain, however, are derived from those of the prolate function expansion and, consequently, only knowledge of the signal over a finite time interval is needed. To be more specific, we arrive at an analytic aproximation to the Fourier transform of the signal and we truncate the Taylor series expansion of the transform. The particular form of the analytic aproximation allows us to bound the L_2 error of the representation both from above and from below. The procedure is structured so that, aside from "known" constant values which must be computed just once for the given class of signals, the only computation required is that of the coefficients in the prolate function expansion of the signal. The finite-record time-domain computation and the particularly simple form of the representing functions are the main features of this method.

II. PRELIMINARIES

A. Notation

Lower case letters will denote functions of time, upper case letters their corresponding Fourier transforms, e. g.

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)e^{j\omega t}d\omega,$$

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}d\omega$$

The bandlimiting operator B and the timelimiting operator T are defined as

$$B \triangleq I_{|\omega|\leqslant\Omega}\,,\quad T \triangleq I_{|t|\leqslant\tau}\,,\quad \Omega,\tau\in\mathbf{R}_+\,,$$

where $I_{(\cdot)}$ is an indicator function in the appropriate domain.

The Hilbert space L_2 with the inner product*

$$<x_1,x_2> = \int_{-\infty}^{\infty} x_1(t)x^*_2(t)\,dt$$

will provide the general background of our discussion; B and T will operate on this space. $R(\cdot)$ will denote the range space of the enclosed operator. $R(B)$, then, is the (Hilbert) space of bandlimited functions of finite power. The L_2 norm will be written as $||\cdot||$, while $||\cdot||_{R(B)}$ will represent the restriction of the L_2 norm to $R(B)$.

Underlying much of the subsequent material is the fact that functions in $R(B)$ are entire in t and, correspondingly, functions in $R(T)$ have Fourier transforms which are entire in ω [1]. When dealing with $x\in R(B)$, we can make use of Pollak's lemma [2,3]

$$|x(t)| \leqslant \left(\frac{\Omega}{\pi}\right)^{1/2} ||x||.$$

(It can be shown that this is a special case of a more general RKHS result [cf. Appendix].)

B. Properties of the Prolates

The bandlimited prolate spheroidal wave functions (PSWFs) ϕ_i are the solutions to the equation [4]

---

* In this paper we will deal only with real functions of time.

$$BT\phi_i = \lambda_i \phi_i \ ,$$

where the eigenvalues λ_i are ordered $(1>\lambda_i>\lambda_{i+1}>0)$ and decrease toward zero rapidly. Analogously, the timelimited PSWFs ψ_i satisfy

$$TB\psi_i = \gamma_i \psi_i \ .$$

It can be shown [5,6] that

$$\frac{B\psi_i}{\sqrt{\gamma_i}} = \phi_i \ , \quad \frac{T\phi_i}{\sqrt{\lambda_i}} = \psi_i \ , \quad \gamma_i = \lambda_i \ .$$

Table I summarizes the corresponding properties of these PSWFs. In particular, properties (2) and (3) illustrate the interesting double orthogonality of the functions. The double orthogonality and completeness in their respective domains make these sets of functions unique [7].

C. Analytic Sampling

In reference [5] an approximate representation for bandlimited signals was derived by using some results due to Slepian [4] and Stuller [19]. To summarize the method we start with an $x \in R(B)$, which we can write as [4]

$$x = \sum_{i=0}^{\infty} x_i \phi_i \ .$$

The timelimited function

$$y \triangleq \sum_{i=0}^{N} \frac{x_i}{\sqrt{\lambda_i}} \psi_i \ ,$$

where N is such that

$$\sum_{i=N+1}^{\infty} x_i^2 < \epsilon \ ,$$

approximates x in the frequency interval $(-\Omega, \Omega)$ to within ϵ L_2 error. Thus, By is a $R(BT)$ approximation to x, with

$$||x - By||^2 < \epsilon \ .$$

Stuller used the appropriate prolate functions to perfectly represent a timelimited function in terms of its uniformly spaced samples taken at one-half the Nyquist sampling rate corresponding to a given bandlimited signal space $R(B)$.

These two concepts were combined and extended to form an approximate representation method for $x \in R(B)$ whereby y, the $R(T)$ approximation to x, was generated à la Slepian; y was unifornly sampled at an arbitrarily slow rate and perfectly reconstructed from these samples via Stuller's scheme; and y was bandlimited to form By, the $R(BT)$ approximation to x. The result was the "nearly perfect" representation of a bandlimited function by means of uniformly spaced samples of a related function taken at an arbitrarily slow rate.

III. DERIVATION OF THE APPROXIMATE REPRESENTATION

A. The Basic Concept

Analysis of the analytic sampling method shows it to be, in a way, wasteful. After the sampling of the time-limited approximation $y(t)$, the reconstruction yields (in the frequency domain, before bandlimitation) $Y(\omega)$ for all ω. Since $Y(\omega)$ approximates the signal transform $X(\omega)$ only for $\omega \in (-\Omega, \Omega)$, what is really needed, and no more, is the accurate reconstruction of $Y(\omega)$ over the interval $(-\Omega, \Omega)$.

Recalling that $Y(\omega)$ is an analytic approximation of $X(\omega)$, we can expand $Y(\omega)$ in a Taylor series about $\omega=0$:

$$Y(\omega) = \sum_{n=0}^{\infty} \frac{Y^{(n)}(0)}{n!} \omega^n \ ,$$

where $Y^{(n)}(0)$ is the n-th derivative of $Y(\omega)$ evaluated at $\omega=0$. Truncation of the series introduces an error term

$$Y(\omega) = \sum_{n=0}^{M} \frac{Y^{(n)}(0)}{n!} \omega^n + R_{M+1}(\omega) \ ,$$

where [8]

$$R_{M+1}(\omega) = \frac{Y^{(M+1)}(\theta\omega)}{(M+1)!} \omega^{M+1} \ ,$$

$$\text{for some } 0<\theta<1 \ .$$

Now, the inverse Fourier transform of $Y^{(n)}(\omega)$ is $(-jt)^n y(t)$, so that

$$Y^{(n)}(0) = \int_{-\infty}^{\infty} (-jt)^n y(t) dt$$

$$= \int_{-T}^{T} (-jt)^n y(t) dt \ , \ \textit{since } y \in R(T).$$

$$\therefore \ Y(\omega) = \sum_{n=0}^{M} \left[\int_{-T}^{T} (-jt)^n y(t) dt \right] \frac{\omega^n}{n!}$$

$$+ R_{M+1}(\omega) \ .$$

We can bound the error $R_{M+1}(\omega)$ by using certain inequalities relating to the Fourier transforms of time-limited functions. Since $Y(\omega)$ is such a Fourier transform*, we can use the Bernstein inequality [9]

$$\sup_{\omega} |Y^{(M+1)}(\omega)| \leqslant \tau^{M+1} \sup_{\omega} |Y(\omega)|$$

along with Pollak's lemma

$$\sup_{\omega} |Y(\omega)| \leqslant \sqrt{2\tau}\ ||y||$$

to give

$$\sup_{\omega} |Y^{(M+1)}(\omega)| \leqslant \sqrt{2\tau}\ \tau^{M+1}\ ||y|| .$$

Thus, we bound the Taylor series truncation error

$$|R_{M+1}(\omega)| \leqslant \sqrt{2\tau} \frac{(\tau\omega)^{M+1}}{(M+1)!}\ ||y|| .$$

Let $\tilde{Y}_M(\omega)$ be the truncation of the Taylor series to M terms, i. e.

$$\tilde{Y}_M(\omega) = \sum_{n=0}^{M} \frac{Y^{(n)}(0)}{n!} \omega^n .$$

The bound on the absolute error

$$|R_{M+1}(\omega)| = |Y(\omega) - \tilde{Y}_M(\omega)|$$

will be used to bound the L_2 error since that is the sense (rather than pointwise) in which $Y(\omega)$ is an approximation of $X(\omega)$.

As discussed previously, we are interested only in the accurate reconstruction of the portion of $Y(\omega)$ in the interval $(-\Omega, \Omega)$. It suffices, therefore, to examine the L_2 error restricted to that interval.

$$\begin{aligned} \int_{-\Omega}^{\Omega} |Y(\omega) - \tilde{Y}_M(\omega)|^2 d\omega &= \\ = \int_{-\Omega}^{\Omega} |R_{M+1}(\omega)|^2 d\omega & \\ \leqslant 2\tau\ ||y||^2 \int_{-\Omega}^{\Omega} \frac{(\tau\omega)^{2(M+1)}}{[(M+1)!]^2} d\omega & \\ = 4\ ||y||^2 \frac{(\tau\Omega)^{2M+3}}{(2M+3)[(M+1)!]^2} . & \end{aligned}$$

or

$$||Y(\omega) - \tilde{Y}_{M-1}(\omega)||^2_{R(B)} \leqslant 4\ ||y||^2 \frac{(\tau\Omega)^{2M+1}}{(2M+1)(M!)^2} .$$

The term on the right hand side which is dependent on M, it should be noted, converges very rapidly to zero with increasing M.

From a theoretical viewpoint the choice of the value of τ seems to be arbitrary; it simply changes the PSWFs used in the formation of $y(t)$. However, we will show in section V that changing τ has an effect on the value of M needed for a given nominal error. Let

$$E_{M-1} \triangleq 4 \left[\frac{(\tau\Omega)^{2M+1}}{(2M+1)(M!)^2} \right]$$

be the bound on the L_2 error of the Taylor series truncation normalized by the energy of $y(t)$. The accompanying table demonstrates the quick convergence to zero of E_{M-1} for various values of $(\tau\Omega)$.

We can put $\tilde{Y}_{M-1}(\omega)$ into a neat form

$$\tilde{Y}_{M-1}(\omega) = \sum_{k=0}^{M-1} \left(\int_{-\tau}^{\tau} t^k y(t)\, dt \right) \frac{(-j\omega)^k}{k!} .$$

So, all that is necessary to produce the approximation $\tilde{Y}_{M-1}(\omega)$ is to calculate the temporal moments of $y(t)$ and form the M-term series. Now, consider

$$\tilde{x}_{M-1}(t) \triangleq B\tilde{y}_{M-1}(t) .$$

Since

$$||Y(\omega) - X(\omega)||^2_{R(B)} < \epsilon$$

and

$$||\tilde{Y}_{M-1}(\omega) - Y(\omega)||^2_{R(B)} \leqslant ||y||^2 E_{M-1} ,$$

we have

$$\begin{aligned} ||X(\omega) - \tilde{X}_{M-1}(\omega)||^2 \leqslant\ & ||X(\omega) - Y(\omega)||^2_{R(B)} \\ & + ||Y(\omega) - \tilde{Y}_{M-1}(\omega)||^2_{R(B)} \\ & + ||\tilde{X}_{M-1}(\omega) - \tilde{Y}_{M-1}(\omega)||^2_{R(B)} \\ & < \epsilon + ||y||^2 E_{M-1} . \end{aligned}$$

$$\therefore ||x - \tilde{x}_{M-1}||^2 < \epsilon + ||y||^2 E_{M-1} ,$$

by Parseval's theorem. A looser but perhaps more explicit bound on the error is

$$||x - \tilde{x}_{M-1}||^2 < \epsilon + ||x||^2 E_{M-1} \lambda^{-1}_{N(\epsilon)} ,$$

which can be easily derived. For large enough M, $\tilde{x}_{M-1}(t)$ is a good bandlimited approximate representation of the original signal $x(t)$. In the general case, though, the representation can only be approximate, whatever the value of M and despite the swift decay of E_{M-1}. This is because $||y||^2$ is a function of ϵ and, in general, $||y||\to\infty$ as $\epsilon\to 0$ [4]. A necessary and sufficient condition for $||y||$ to stay finite as $\epsilon\to 0$ is $x\in R(BT)$ [4,10]. For this class of bandlimited signals the last inequality becomes

---

* Formally it is an analytic function of exponential type $\leqslant\tau$ [1].

$$||x-\tilde{x}_{M-1}||^2 \leqslant ||y||^2\, E_{M-1}$$

and, in the limit, the representation will be perfect. If x is "essentially timelimited", then $||y||^2$ will be relatively small and a small value of N should give a reasonably faithful representation of x.

B. An Explicit Formula for $\tilde{x}_{M-1}(t)$

An explicit formula for $\tilde{x}_{M-1}(t)$ can be derived. By definition $\tilde{x}_{M-1}(t)$ is in $R(B)$ and

$$\tilde{X}_{M-1}(\omega) = \sum_{k=0}^{M-1}\left[\int_{-\tau}^{\tau} t^k y(t)\,dt\right]\frac{(-j\omega)^k}{k!}\; I_{|\omega|\leqslant\Omega}$$

Transforming, we have

$$\tilde{x}_{M-1}(t) =$$

$$= \sum_{k=0}^{M-1}\left[\int_{-\tau}^{\tau} t^k y(t)\,dt\right] F^{-1}\left[\frac{(-j\omega)^k}{k!} I_{|\omega|\leqslant\Omega}\right] .$$

Some simple calculations yield

$$F^{-1}\left[\frac{(-j\omega)^k}{k!}\; I_{|\omega|\leqslant\Omega}\right] = \frac{(-1)^k}{k!}\frac{d^k}{dt^k}\left(\frac{\sin\Omega t}{\pi t}\right)$$

Leibniz's rule on the n-th derivative of a product [11] can then be used to derive the following equation.

$$\frac{d^k}{dt^k}\left(\frac{\sin\Omega t}{\pi t}\right) = \sum_{i=0}^{k}\left[\binom{k}{i}\frac{(-1)^i}{\pi}\cdot\right.$$

$$\left.\cdot\,(i)!\,\Omega^{(k-i)}\right]\left[t^{-(i+1)}\sin\left(\Omega t+(k-i)\frac{\pi}{2}\right)\right] .$$

Substituting this into the expression for $\tilde{x}_{M-1}$, we have

$$\tilde{x}_{M-1}(t) =$$

$$= \frac{1}{\pi}\sum_{k=0}^{M-1} m_k \sum_{i=0}^{k} c_{i,k}\, t^{-(i+1)}\sin\left[\Omega t+(k-i)\frac{\pi}{2}\right]$$

where

$$m_k \triangleq \int_{-\tau}^{\tau} t^k y(t)\,dt$$

and

$$c_{i,k} \triangleq \frac{(-1)^{k+i}}{(k-i)!}\Omega^{(k-i)} .$$

If we let

$$d_k(\Omega,t) \triangleq \sum_{i=0}^{k} c_{i,k}\, t^{-(i+1)}\sin\left[\Omega t+(k-i)\frac{\pi}{2}\right] ,$$

then the formula for $\tilde{x}_{M-1}(t)$ can be written as

$$\tilde{x}_{M-1}(t) = \frac{1}{\pi}\sum_{k=0}^{M-1} m_k d_k(\Omega,t) .$$

The $d_k(\Omega,t)$'s are known finite sums of elementary functions independent of the particular signal but dependent on the frequency band of interest. The m_k's are the temporal moments of the timelimited approximation to the particular signal x.

Since $||y||\to\infty$ as $\epsilon\to 0$, it is to be expected that the m_k's will increase in absolute value as ϵ decreases. For a signal x with most of its energy outside the interval $(-\tau,\tau)$, the m_k coefficients will also be quite large and their numerical computation might pose serious problems, particularly in a noisy environment. For the class of signals in $R(BT)$ with much of their energy in the interval $(-\tau,\tau)$, however, the method seems to be neat and practical. If the scheme is shown to work in practice for $R(BT)$ signals, the density of $R(BT)$ in $R(B)$ will make it practicable to use on all the signals in $R(B)$[10]. For the moment, let's just note that the computation of the m_k's is the only difficult part of this scheme; the elementary functions can be generated quite easily. It is of interest, then, to look at a way of evaluating the m_k coefficients directly from the timelimited version Tx of the signal.

C. Evaluation of the Temporal Moments

The bandlimited signal x can be represented in terms of the PSWFs ψ_i [5]

$$x(t) = \sum_{i=0}^{\infty} x_i\phi_i(t) = \sum_{i=0}^{\infty}\frac{x_i}{\sqrt{\lambda_i}}B\psi_i(t)$$

where

$$x_i = \frac{1}{\lambda_i}\int_{-\tau}^{\tau} x(t)\phi_i(t)\,dt .$$

The timelimited approximation $y(t)$ is an implicit function of ϵ:

$$y(t) = \sum_{i=0}^{N(\epsilon)}\frac{x_i}{\sqrt{\lambda_i}}\psi_i(t) \in R(T)$$

and

$$||x-y||^2_{R(B)} = \sum_{i=N+1}^{\infty} x_i^2 .$$

The number N is chosen so that

$$||x-y||^2_{R(B)} < \epsilon .$$

Now,

$$m_k \triangleq \int_{-\tau}^{\tau} t^k y(t)\,dt = \sum_{i=0}^{N} x_i \int_{-\tau}^{\tau} t^k \frac{\psi(t)}{\sqrt{\lambda_i}}\,dt$$

or

$$m_k = \sum_{i=0}^{N} x_i n_{i,k} ,$$

where we let

$$n_{i,k} \triangleq \lambda_i^{-1/2} \int_{-\tau}^{\tau} t^k \psi_i(t)\,dt .$$

The PSWF coefficients $\{x_i\}$ are computed from the signal over the known interval $(-\tau,\tau)$. The $n_{i,k}$'s are the known, scaled (by $\lambda_i^{-1/2}$) temporal moments of the PSWFs $\{\psi_i\}$. They can be calculated beforehand and stored. All calculations are made on the finite time interval $(-\tau,\tau)$. This desirable feature is a result of the nice properties of the PSWFs outlined in section II.

If, besides the time-truncated signal Tx, we are given the energy $||x||^2$ of the original signal, then we have a way of finding the desired N which is easier than calculating the x_i for $i=0,1,2,\ldots,K$ (K large)* all at once and using

$$\sum_{i=M+1}^{K} x_i^2 < \epsilon$$

to find M.

One of the main features of our representation is the particularly simple form of the $d_k(\Omega,t)$ functions. Aside from some known constants, these consist of a finite sum of elementary functions: inverse powers of t and sines and cosines. The question which naturally arises is "Are the m_k's the best ceofficients for representing $x(t)$ in terms of these elementary functions, and, if not, how can we make them better?". This is answered in the succeeding section.

IV. RELATION TO THE OPTIMAL REPRESENTATION BY ELEMENTARY FUNCTIONS

Gallagher and Wise [3] have proposed a representation for bandlimited functions which consists of expanding the Fourier transform, say $X(\omega)$, in terms of the Legendre polynomials $P_n(\omega)$, so that

$$X(\omega) = \sum_{n=0}^{\infty}\left[(n+\tfrac{1}{2})\int_{-\Omega}^{\Omega} X(\omega)P_n(\frac{\omega}{\Omega})\,d\omega\right]P_n(\frac{\omega}{\Omega})$$

where, as is customary, we define the n-th Legendre polynomial by $P_0(\omega)=1$ and Rodrigues' formula [12]

$$P_n(\omega) = \frac{1}{2^n n!}\frac{d^n(\omega^2-1)}{d\omega^n} , \quad n=1,2,3,\cdots$$

Taking the inverse Fourier transform, we have the representation

$$x(t) = \frac{\Omega}{\pi}\sum_{n=0}^{\infty}\left[(j)^n(n+\tfrac{1}{2})\cdot \cdot\int_{-\Omega}^{\Omega} X(\omega)P_n(\frac{\omega}{\Omega})\,d\omega\right] j_n(\Omega t) ,$$

or

$$x(t) = \frac{\Omega^2}{\pi^2}\sum_{n=0}^{\infty}\left[(-1)^n(n+\tfrac{1}{2})\cdot \cdot\int_{-\infty}^{\infty} x(t)j_n(\Omega t)\,dt\right] j_n(\Omega t) .$$

Here $j_n(t)$ is the n-th spherical Bessel function of the first kind [12]. As noted by Gallagher and Wise, the spherical Bessel function can be represented by a power series as

$$j_n(t) = \frac{\pi^{1/2}}{2}\left(\frac{t}{2}\right)^n \sum_{k=0}^{\infty} \frac{(-1)^k t^{2k}}{4^k k!\,\Gamma(n+k+\frac{3}{2})} .$$

There are also representations of $j_n(t)$ by elementary functions [12]

$$j_n(t) = f_n(t)\sin t + (-1)^{n+1} f_{-n-1}(t)\cos t ,$$

with the coefficients being generated by the recursion

$$f_0(t) = t^{-1} \quad , \quad f_1(t) = t^{-2}$$

$$f_{n-1}(t) + f_{n+1}(t) = (2n+1)t^{-1} f_n(t) , \quad n \in \mathbf{Z}.$$

---

* For a signal with much of its energy in the interval $(-\tau,\tau)$, K would not have to be very large for $\sum_{0}^{K} x_i\phi_i(t)$ to be a good approximation to x(t).

These elementary functions are precisely those which occur in our representation of bandlimited signals. The reason for this is easy to see. An n-term Taylor series expansion in frequency is simply an n-th order polynomial approximation of the bandlimited spectrum, as is the n-term Legendre polynomial expansion proposed by Gallagher and Wise; the difference being, of course, the coefficients of the respective polynomials. It is known that least squares approximation of $X(\omega)$ by a polynomial of degree n in ω is attained for one and only one set of polynomial coeffients [13]. This n-th order polynomial is precisely the n-term Legendre polynomial expansion of $X(\omega)$.

As was shown in the previous section, the Taylor series expansion we take is that of $BY(\omega)$, the $R(BT)$ approximation to $X(\omega)$. If we denote the M-term Legendre polynomial expansion of $X(\omega)$ by $L_{M-1}(X)$, and its Fourier transform by $l_{M-1}(x)$, we get the following lower bounds on the final error of our approximation.

$$||x-l_{M-1}(x)||^2 \leqslant ||x-l_{M-1}(By)||^2$$
$$\leqslant ||x-\tilde{x}_{M-1}||^2$$

The secondary lower bound is

$$||x-l_{M-1}(x)||^2 = ||X-L_{M-1}(X)||^2$$
$$= \frac{1}{2\pi}\sum_{n=M}^{\infty}(n+\tfrac{1}{2})\left|\int_{-\Omega}^{\Omega} X(\omega)P_n(\frac{\omega}{\Omega})d\omega\right|^2 .$$

The primary lower bound

$$||x-l_{M-1}(By)||^2$$

is of similar form, with $X(\omega)$ replaced by the approximation $BY(\omega)$. Since

$$\tilde{X}_{M-1}(\omega) = \sum_{n=0}^{M-1}\frac{Y^{(n)}(0)}{n!}\, I_{|\omega|\leqslant\Omega} ,$$

the approximation $\tilde{X}_{M-1}(\omega)$ may be viewed (trivially) as the best L_2 approximation of the M-term Taylor series of $BY(\omega)$ by an $(M-1)$-degree polynomial. In this sense, this is the best we can do, given the data $\{Y^{(n)}(0)\}_{n=0}^{M-1}$. By construction we have $||x-By||^2 < \epsilon$, so that the two lower bounds should be quite close to each other. For the special case of $X(\omega)$ having a simple polynomial shape, the bounds would be nearly zero, and the approximation $\tilde{x}_{M-1}$ would be accurate for a relatively small value of M. For $X(\omega)$ of significantly complex shape⊛ the value of M for any great accuracy of approximation would have to be large.

The derivation of a simple estimate of M for a desired upper bound on the error is the subject of the next section.

V. A SIMPLE ESTIMATE OF M

A bound on the L_2 error of the Taylor series truncation (normalized by $||y||^2$) was shown in section III to be

$$E_{M-1} = \frac{4(\tau\Omega)^{2M+1}}{(2M+1)(M!)^2} .$$

This error bound can be tabulated as a function of τ and M, allowing us to choose an appropriate M for a desired E_M, once we are given τ. It is of interest, however, to have a simple formula for an estimate of M, say $\hat{M}(\tau,E_M)$, which would yield $E_{\hat{M}}\approx E_M$. The following is a heuristic derivation of an essentially empirical formula for just such an estimate.

Stirling's approximation for $M!$ is good for $M>9$ [14] and, therefore, it may be used in the above equation. Thus,

$$E_{M-1} \approx \frac{1}{\pi e(M+\tfrac{1}{2})}\left(\frac{\tau\Omega e}{M}\right)^{2M+1} . \quad (*)$$

Ignoring the ½ and rearranging,

$$E_{M-1} \approx \frac{1}{\pi e M^2}\left(1-\frac{2(M-\tau\Omega e)}{2M}\right)^{2M} .$$

Taking natural logarithms,

$$\ln M + \ln\sqrt{\pi e} + \ln\sqrt{E_{M-1}} \approx$$
$$\approx M \ln\left(1-\frac{2(M-\tau\Omega e)}{2M}\right) .$$

The approximation $\ln(1+x)\approx x$ may be used to get

$$M + \ln M \approx \tau\Omega e - \ln\sqrt{\pi e} - \ln\sqrt{E_{M-1}} .$$

Assuming $M=50$⧆, then $\ln M \approx 3.9$, so

⊛ We may say $X(\omega)$ has a "significantly complex shape" if every close (in L_2 norm) polynomial approximation of $X(\omega)$ is of high degree.

⧆ If the true M is in the range 10-1000, then this assumption results in an estimate M which is within 3 of the true value.

$$\hat{M} = \left\lceil \tau\Omega e - 6 - \ln\sqrt{E_M} \right\rceil , \qquad (**)$$

where $\lceil x \rceil$ denotes the smallest integer greater than or equal to x. The accompanying table of $\hat{M}(\tau\Omega, E_M)$ and $E_{\hat{M}}$ demonstrates the accuracy of this simple estimate.

It is clear from examining equation (*) that we need $M \geqslant \tau\Omega e$ for a small error bound. Moreover, both that equation and equation (**) indicate that asymptotically, as $\tau \rightarrow \infty$, $M = \tau\Omega e$ gives an error bound E_M which is arbitrarily small. Recall from section III that the bound for the total error of our M-th order representation is

$$\epsilon + ||y||^2 E_M .$$

For any given $\epsilon > 0$, $||y||^2$ is finite. As a consequence, this method is able to represent a bandlimited signal to within an arbitrarily small error by $M = \tau\Omega e$ simple functions, i.e. derivatives of $\frac{\sin\Omega t}{\pi t}$, asymptotically as $\tau \rightarrow \infty$. We can compare this with a known result concerning the dimension of the space of bandlimited signals.

Slepian's version [15] of the Landau-Pollak theorem [16] states that the approximate asymptotic dimension of $R(B)$ as $\tau \rightarrow \infty$ is $\frac{2}{\pi}(\tau\Omega)$. This number is achieved only by expanding the signal in terms of the PSWFs. We expect any other series expansion to have an asymptotic number which is greater than $\frac{2}{\pi}(\tau\Omega)$. We will call the ratio of the approximate asymptotic dimension of $R(B)$ to the asymptotic number of simple functions our method needs to approximately cover $R(B)$ the relative asymptotic efficiency or RAE. Thus,

$$RAE = \frac{\frac{2}{\pi}\tau\Omega}{\tau\Omega e} = \frac{2}{\pi e} \approx .23$$

This representation method, then, is one fourth as asymptotically efficient as the optimal representation (by PSWFs). However, the functions involved in the final reconstruction of the original signal are much simpler to generate than the PSWFs.

VI. CONCLUSION

A novel scheme for the approximate representation of bandlimited signals by sums of elementary functions was presented, and some basic theoretical considerations were discussed. The procedure shares many of the advantages and disadvantages of a prolate spheroidal wave function representation, such as: adequacy of finite data length, fast decay of representing functions, and difficulty in the generation of coefficients. It was demonstrated to be asymptotically one-fourth as efficient as the latter, but with a final reconstruction which is of much simpler form.

Numerical evaluation of this method, particularly in a digital framework [20], will be necessary to confirm its usefulness. Also, a simple empirical or analytical formula for estimating $N(\epsilon)$ as the size of the known interval varies would nicely complement the estimate of M we found. Lastly, the quick decay and the bandlimitation of the elementary functions used in our procedure might find application in a bandwidth-preserving scrambling scheme similar to that proposed by Wyner [21].

APPENDIX

Aronszajn [22] developed the following result: Given a reproducing kernel Hilbert space with kernel $K(\cdot,\cdot)$, and an orthonormal system $\{g_n\}$ in the $RKHS$, then, for any l_2 sequence $\{\alpha_n\}$,

$$\sum_0^\infty |\alpha_n||g_n(x)| \leqslant K(x,x)^{1/2} \left[\sum_0^\infty |\alpha_n|^2\right]^{1/2} .$$

As a special case we identify

$$K(t,s) \equiv \frac{\sin\Omega(t-s)}{\pi(t-s)} , \quad g_n(t) \equiv \phi_n(t) ,$$

$$RKHS \equiv R(B) .$$

Then,

$$K(t,t) = \frac{\sin\Omega t}{\pi t}\Big|_{t=0} = \frac{\Omega}{\pi} ,$$

and

$$\sum_0^\infty |\alpha_n||\phi_n(t)| \leqslant \left[\frac{\Omega}{\pi}\right]^{1/2} \left[\sum_0^\infty |\alpha_n|^2\right]^{1/2} . \qquad (A)$$

This is a generalization of Pollak's lemma, as is easily shown below. For any x in $R(B)$,

$$\sum_0^\infty x_n^2 = ||x||^2 < \infty \quad (i.\ e.\ \{x_n\} \in l_2)$$

and it follows from equation (A) that

$$|x(t)| = |\sum_0^\infty x_i \phi_i(t)|$$

$$\leqslant \sum_0^\infty |x_i||\phi_i(t)| \leqslant \left(\frac{\Omega}{\pi}\right)^{1/2} \left(\sum_0^\infty x_i^2\right)^{1/2}$$

or

$$|x(t)| \leqslant \left(\frac{\Omega}{\pi}\right)^{1/2} ||x||\,,$$

which is Pollak's lemma.

REFERENCES

[1] R. E. A. C. Paley and N. Wiener, *Fourier Transforms in the Complex Domain* , New York: Amer. Math. Soc., 1934, p.13.

[2] H. J. Landau and W. L. Miranker, "The Recovery of Distorted Band-Limited Signals," Jour. Math. Anal. and Appl., vol. 2, pp. 97-104 (1961).

[3] N. C. Gallagher, Jr. and G. L. Wise, "A Representation for Band-Limited Functions," Proc. IEEE, vo. 63, pp. 1624-1625, Nov. 1975.

[4] D. Slepian and H. O. Pollak, "Prolate Spheroidal Wave Functions -I," Bell. Sys. Tech. Jour., pp. 43-65, Jan. 1961.

[5] A. Sonnenschein and B. W. Dickinson, "On the Representation of a Bandlimited Signal -I," Proc. 1979 Conf. on Inf. Sci. and Sys., pp. 247-253.

[6] A. K. Jain and S. Ranganath, "Extrapolation and Spectral Estimation Techniques for Discrete Time Signals," Univ. Calif. at Davis, Technical Report SIPL-78-1, Nov. 1978.

[7] I. F. Krasichkov, "Systems of functions with the double orthogonality property," Matematicheskie Zametki, vol. 4, No. 5 (1968), pp. 551-556. (In Russian)

[8] K. Rectorys, *Survey of Applicable Mathematics,* Cambridge: MIT Press, 1969, p. 435.

[9] N. I. Achieser, *Theory of Approximation,* New York: Ungar, 1956, p.138 .

[10] A. Sonnenschein and B. W. Dickinson, "Extending the Applicability of an Extrapolation Procedure for Bandlimited Signals," submitted for publication.

[11] I. S. Gradshteyn and I. M. Ryzhik, *Tables of Integrals, Series, and Products,* New York: Academic, 1965, p. 19 .

[12] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions,* New York: Dover, 1972.

[13] D. Jackson, *The Theory of Approximation,* New York: Amer. Math. Soc., 1930, pp. 95-96.

[14] W. Feller, *An Introduction to Probability Theory and Its Applications,* New York: Wiley, 1968, p.52 .

[15] D. Slepian, "On Bandwidth," Proc. IEEE, vol. 64, pp. 292-300, Mar. 1976.

[16] H. J. Landau and H. O. Pollak, "Prolate Spheroidal Wave Functions -II," Bell Syst. Tech. Jour., pp. 65-84, Jan. 1961 .

[17] K. Yao, "On Some Representations and Sampling Expansions for Band-Limited Signals," doctoral dissertation, Princeton University, Princeton, N. J. April 1965.

[18] C. E. Shannon, "A Mathematical Theory of Communication," Bell Syst. Tech. Jour., vol. 27, 1948, pp.379-423, 623-659.

[19] J. A. Stuller, "Representation of Finite Duration Signals," IEEE Trans. on Inf. Theory, vol. IT-18, pp.667-669, Sept. 1972.

[20] D. Slepian, "Prolate Spheroidal Wave Functions- V: The Discrete Case," Bell Syst. Tech. Jour., pp.1371-1431, May-June 1978.

[21] A. Wyner, "An Analog Scrambling Scheme which Does Not Expand Bandwidth, Part I: Discrete Time," IEEE Trans. on Inf. Theory, vol. IT-25, No.3, pp.261-274, May 1979.

[22] N. Aronszajn, "Theory of Reproducing Kernels," Trans. Amer. Math. Soc., vol. 63, No. 3, pp. 337-404, May 1950.

[23] H. Landau, "Sampling, Data Transmission, and the Nyquist Rate," Proc. IEEE, vol. 55, No. 10, pp.1701-1706, Oct. 1967.

TABLE I. SYMMETRIC PROPERTIES OF THE PSWFs ϕ_i AND ψ_i

1.	$BT\phi_i = \lambda_i\phi_i$	$TB\psi_i = \gamma_i\psi_i$
2.	$< \phi_i, \phi_j > = \delta_{ij}$	$< \psi_i, \psi_j > = \delta_{ij}$
3.	$< T\phi_i, T\phi_j > = \lambda_i\delta_{ij}$	$< B\psi_i, B\psi_j > = \gamma_i\delta_{ij}$
4.	$1 > \lambda_i > \lambda_{i+1} > 0$	$1 > \gamma_i > \gamma_{i+1} > 0$
5.	$\lambda_i \rightarrow 0$ as $i \rightarrow \infty$	$\gamma_i \rightarrow 0$ as $i \rightarrow \infty$
6.	$\{\phi_i\} - CON$ set in $R(B)$	$\{\psi_i\} - CON$ set in $R(T)$
7.	$\{T\phi_i\} - CON$ set in $R(T)$	$\{B\psi_i\} - CON$ set in $R(B)$

TABLE II. CONVERGENCE OF E_M

$\tau\Omega$	M	$\log_{10}E_M$
1	5	-6.2
	6	-8.0
	7	-9.8
10	30	-6.0
	32	-8.1
	34	-10.3
100	274	-6.2
	276	-7.9
	277	-10.6

TABLE III. COMPARISON OF $E_{\hat{M}}$ AND E_M

$\tau\Omega$	$\log_{10}E_M$	$\hat{M}$	$\log_{10}E_{\hat{M}}$
10	-6	29	-5.0
	-8	31	-7.0
	-10	33	-9.3
100	-6	273	-5.7
	-8	276	-7.9
	-10	278	-9.7
1000	-6	2720	-6.7
	-8	2722	-8.5
	-10	2724	-10.2

FIXED-POINT SUBOPTIMAL SMOOTHING

DONALD O. BENSON, Jr.
Dynamics Research Corporation
Wilmington, Massachusetts 01887

ABSTRACT

A general solution to suboptimal linear smoothing is presented based on fixed-point smoothing. Previous works on suboptimal smoothing have analyzed the effects of errors in initial conditions, P_o, driving noise, Q, and measurement noise, R. Solutions for errors in the measurement matrix, H, and system dynamics, ϕ or F, are available for special cases, and these solutions are complicated compared to filtering algorithms.

The suboptimal smoothing algorithm derived here is based on suboptimal filtering results and is easily implemented. The technique is especially suited for cases where the estimation error covariance of the entire state vector is not required at every time point.

I. INTRODUCTION

Suboptimal smoother designs generally do not receive the extensive error analyses compared to suboptimal filter designs. Apart from the fact that filter applications are more numerous than smoother applications, a good portion of the problem stems from the formidable appearance of the available suboptimal smoothing equations as well as the lack of a solution to the problem of structure and dimension mismatch between smoother and actual system dynamics [1]-[3].

The method presented in this paper alleviates and solves these problems by deriving an analysis of suboptimal smoothing based on fixed-point techniques. This solution should make analysis of suboptimal smoothers much more accessible to the practicing engineer since the form of the equations is based on the widely understood and easily implemented suboptimal filtering equations.

In the design and analysis of suboptimal smoothers questions identical to suboptimal filter applications arise. These include:

1. The designer may have a well defined mathematical model for the system and wants to design and analyze reduced state smoothers that will reduce computational complexity and costs.

2. The designer does not have complete knowledge of the actual system model, and he wants to evaluate the sensitivity of a smoother design to variations in the system parameters and/or structure.

3. The designer may want to design and analyze smoother performance when both smoother and actual model dimensions and parameters are varied as in, for example, the suboptimal filter applications of the minimax criteria described in [4, 5].

Previous analyses [1, 2] of fixed interval suboptimal smoothers have derived solutions for smoothing error covariance due to initial condition covariance, P_o, driving noise covariance, Q, and measurement noise covariance, R. Solutions for errors in the measurement matrix, H, and system dynamics ϕ or F, are available for the special cases of suboptimal smoothers derived for identical system and estimator state variables or by eliminating states from the model. While these solutions yield estimation error covariances for the whole state vector at every time point, they are complicated to implement compared to the fixed-point suboptimal smoothing solution presented in this paper. A sensitivity analysis of fixed-point smoothing is presented in [3]; however, their solution does not use the filter form for the suboptimal smoothing problem, and their solution is computationally inefficient since it applies to the entire state vector rather than to only the states or linear combinations of the states of interest. Furthermore, their solution applies to the case of parameter changes with identical definitions of the system and smoother state variables. The problem of different dimension and model structure between the system and smoother does not seem to be addressed. A smoothing sensitivity analysis for a finite memory smoother for a time-invariant with a constant sampling interval has been derived in [6]. Work on optimal fixed-point smoothing, for example [7, 8], derive the optimal solution for the entire state vector. This paper shows that optional fixed-point smoothing results can be obtained easily for any linear combination of the state vector.

The features of the suboptimal smoothing solution presented in this paper then are threefold: (1) suboptimal filtering algorithms can be used directly for fixed-point suboptimal smoothing, (2) different structure and dimension between system and smoother models are analyzed, and (3) the smoothed states can be any linear combination of the states of interest - not necessarily the entire state vector.

The solution presented here is the discrete measurement case, and the analysis can be extended easily for continuous measurements. Section II presents the general fixed-point suboptimal filter results, while Section III is a special case often encountered in engineering applications. The special case solution allows decreasing the state size, thus easing the computational burden. A numerical example is presented in Section IV.

II. FIXED-POINT SUBOPTIMAL SMOOTHING - GENERAL CASE

The approach taken in this section is to derive the gains for the suboptimal smoother based on fixed-point techniques. Once the gains have been obtained, then the suboptimal smoother error analysis is similar to that used for suboptimal filters except that the state vector is augmented at the times smoothed estimation error covariances are desired.

A. Determining the Smoothing Gains

The fixed-point smoothing method uses the filter gains and estimates until the time the smoothing gains and estimates are desired. At this time the state is augmented by the linear combination of the states to be smoothed. The dynamics for the augmented system are used along with the appropriate initial conditions, to compute the fixed-point smoothing gains for the remaining time interval.

The model for the system dynamics and measurement used for the smoother design before the smoothing times of interest are

$$X^*_k = \phi^* X^*_{k-1} + W^*_{k-1} \qquad \text{(system)} \qquad (1)$$

$$Z_k = h^* X^*_k + V^*_k \qquad \text{(measurement)} \qquad (2)$$

with initial condition covariance P^*_o, driving noise covariance Q^*, measurement noise covariance R^*, (note that ϕ^*, h^*, Q^*, R^* can be functions of k).

At time, t_s, the smoothed estimate of a linear combination of the states is desired. Note that t_s need not be a measurement time. This is accomplished by letting the states of interest be

$$Y_s = L^* X^*_s \qquad (3)$$

Notice that Y_s is not necessarily the entire state vector. The dynamics of Y_s is easily determined by observing that the value of the state in past time is a constant. Thus

$$\dot{Y}_s = 0 \quad \text{for } t > t_s \qquad (4)$$

and $\hat{Y}_k$ will be smoothed estimate of $L^* X^*_s$ at t_s given k measurements. The gain sequence, K^*_k, is determined from the Kalman estimation error covariance solution of (1) and (2) for $t < t_s$. At $t = t_s$ the state vector is augmented by Y_s and the covariance at this time is:

$$P^*_{A_s} = \left[\begin{array}{c} I \\ \hline L^* \end{array}\right] P^*_s \left[\begin{array}{c} I \\ \hline L^* \end{array}\right]^T \qquad (5)$$

where T denotes transpose, I is identity, and P^*_s is the estimation error covariance of $\hat{X}^*$ at $t = t_s$. If t_s is a measurement time the effect of that measurement can be included either before or after the state is augmented. For $t > t_s$ the system and measurement become:

$$\begin{bmatrix} X^* \\ Y \end{bmatrix}_k = \begin{bmatrix} \phi^* & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} X^* \\ Y \end{bmatrix}_{k-1} + \begin{bmatrix} W^*_{k-1} \\ 0 \end{bmatrix} \qquad \text{(system)} \qquad (6)$$

$$Z_k = [h^* \vdots 0] \begin{bmatrix} X^* \\ Y \end{bmatrix}_k + V^*_k \qquad \text{(measurement)} \qquad (7)$$

For $t > t_s$ the gain sequence for the augmented solution is determined from the Kalman estimation error covariance solution of (6) and (7) with condition (5) at $t = t_s$. Notice that there is no coupling between X^* and Y in the dynamics (6) or the measurement (7). The fixed-point smoothing is accomplished via correlations established in the Kalman filter covariance and gain calculations with the initial correlation given by (5).

As developed in this section, the gains for the fixed-point smoother are determined by (1-7). While these gains and corresponding estimates are optimal for the design states, X^*, dynamics, ϕ^*, etc., and the filter/smoother analysis of its own performance is P^*_k, the actual filter/smoother performance is suboptimal since the actual state is X, the actual dynamics is ϕ, etc.

The resulting error analysis will be discussed next.

B. Error Analysis

Once the gains, K_k^*, have been obtained the suboptimal fixed-point error analysis proceeds as follows.[1] The estimate is propagated and updated as:

$$\hat{X}_k^{*-} = \phi^* \hat{X}_{k-1}^{*+} \tag{8}$$

$$\hat{X}_k^{*+} = \hat{X}_k^{*-} + K_k^* (Z_k - h^* \hat{X}_k^{*-}) \tag{9}$$

where + and - denote after and before a measurement. For times $t_k > t_s$ the augmented state vector and corresponding gain, $K_{A_k}^*$, are used in (8, 9).

The actual system and measurements are defined by:

$$X_k = \phi X_{k-1} + W_{k-1} \tag{10}$$

$$Z_k = h X_k + V_k \tag{11}$$

with the state vector augmented by

$$Y_s = L X_s \tag{12}$$

at time $t = t_s$.

Combine the filter (8, 9) with the system (10, 11) to yield the augmented estimate/system propagate and update state equations, viz.:

$$\begin{bmatrix} \hat{X}^* \\ X \end{bmatrix}_k^- = \begin{bmatrix} \phi^* & 0 \\ 0 & \phi \end{bmatrix} \begin{bmatrix} \hat{X}^* \\ X \end{bmatrix}_{k-1}^+ + \begin{bmatrix} 0 \\ W \end{bmatrix}_{k-1} \tag{13}$$

$$\begin{bmatrix} \hat{X}^* \\ X \end{bmatrix}_k^+ = \begin{bmatrix} \hat{X}^* \\ X \end{bmatrix}_k^- - \begin{bmatrix} K_k^* \\ \hdashline 0 \end{bmatrix} \left\{ [h^* \vdots -h] \begin{bmatrix} \hat{X}^* \\ X \end{bmatrix}_k^- - V_k \right\} \tag{14}$$

The covariance of (13) and (14) for $t < t_s$ are given by

$$P_k^- = \Phi P_{k-1}^+ \Phi^T + Q_{k-1} \tag{15}$$

$$P_k^+ = (I - K_k H) P_k^- (I - K_k H)^T + K_k R_k K_k^T \tag{16}$$

with initial condition

$$P = \begin{bmatrix} 0 & 0 \\ 0 & P_o \end{bmatrix}, \quad \text{and } Q_k = \begin{bmatrix} 0 & 0 \\ 0 & q_k \end{bmatrix}, \quad \Psi = \begin{bmatrix} \phi^* & 0 \\ 0 & \phi \end{bmatrix}, \quad K_k = \begin{bmatrix} K_k^* \\ \hdashline 0 \end{bmatrix},$$

$$H = [h^* \vdots -h] \tag{17}$$

At $t = t_s$ augment (13) and (14) with the error in the smoothed states determined from (3) and (12) as:

$$e_s = \hat{Y}_s - Y_s = L^* \hat{X}_s^* - L X_s \tag{18}$$

---

[1] Although (1-7) represent a convenient way to determine the gains, gain sequences computed by other methods can be used for the suboptimal smoothing error analysis.

Equations corresponding to (13) and (14) can be written:

$$\begin{bmatrix}\hat{X}^* \\ e_s \\ X\end{bmatrix}_k^- = \begin{bmatrix}\phi^* & 0 & 0 \\ 0 & I & 0 \\ & 0 & \phi\end{bmatrix}\begin{bmatrix}\hat{X}^* \\ e_s \\ X\end{bmatrix}_{k-1}^+ + \begin{bmatrix}0 \\ 0 \\ W\end{bmatrix}_{k-1} \tag{19}$$

$$\begin{bmatrix}\hat{X}^* \\ e_s \\ X\end{bmatrix}_k^+ = \begin{bmatrix}\hat{X}^* \\ e_s \\ X\end{bmatrix}_k^- - \left[\begin{array}{c}K^*_{A_k} \\ \hline 0\end{array}\right]\left\{[h^* \vdots 0 \vdots -h]\begin{bmatrix}\hat{X}^* \\ e_s \\ X\end{bmatrix}_k^- - V_k\right\} \tag{20}$$

and covariance at $t = t_s$ determined by

$$P_{A_s} = \left[\begin{array}{c|c}I_{\hat{x}} & 0 \\ \hline L^* & -L \\ \hline 0 & I_x\end{array}\right] P_s \left[\begin{array}{c|c}I_{\hat{x}} & 0 \\ \hline L^* & -L \\ \hline 0 & I_x\end{array}\right]^T \tag{21}$$

where P_s is the covariance of $\begin{bmatrix}\hat{X}^* \\ X\end{bmatrix}$ at time, t_s. After t_s the covariances are determined by (15) and (16) for the augmented system where

$$Q_k = \begin{bmatrix}0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & q_k\end{bmatrix}, \quad \Phi = \begin{bmatrix}\phi^* & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & \phi\end{bmatrix}, \quad K_k = \left[\begin{array}{c}K^*_{Ak} \\ \hline 0\end{array}\right], \quad \text{and}$$

$$H = [h^* \vdots 0 \vdots -h] \tag{22}$$

A general solution to the fixed-point suboptimal smoothing problem is then determined by (15-17) and (21-22).

III. STATE SIZE REDUCTION

In many engineering applications part of the filter dynamics and system are identical. The error state vector then can be used for covariance calculations resulting in decreased state vector size and considerable computational savings. This is done by partitioning the filter and system as:

$$\begin{bmatrix}\hat{X}^*_1 \\ \hat{X}^*_2\end{bmatrix}_k^- = \begin{bmatrix}\phi^*_{11} & \phi^*_{12} \\ 0 & \phi^*_{22}\end{bmatrix}\begin{bmatrix}\hat{X}^*_1 \\ \hat{X}^*_2\end{bmatrix}_{k-1}^+ \tag{23}$$

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix}_k = \begin{bmatrix} \phi_{11} & \phi_{12} \\ 0 & \phi_{22} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}_{k-1} + \begin{bmatrix} W_1 \\ W_2 \end{bmatrix}_{k-1} \qquad (24)$$

where $\phi_{11} = \phi^*_{11}$, and the dimension of $\hat{X}^*_2$ does not necessarily equal the dimension of X_2. Forcing functions for the system and filter are generated by X_2 and $\hat{X}^*_2$ so that $\phi^*_{21} = 0$ and $\phi_{21} = 0$. Define $e = \hat{X}^*_1 - X_1$, then filter/system propagation corresponding to (13) or (19) is:

$$\begin{bmatrix} e \\ \hat{X}^*_2 \\ X_2 \end{bmatrix}^- = \begin{bmatrix} \phi_{11} & \phi^*_{12} & -\phi_{12} \\ 0 & \phi^*_{22} & 0 \\ 0 & 0 & \phi_{22} \end{bmatrix} \begin{bmatrix} e \\ \hat{X}^*_2 \\ X_2 \end{bmatrix}^+_{k-1} \begin{bmatrix} -W_1 \\ 0 \\ W_2 \end{bmatrix}_{k-1} \qquad (25)$$

If the measurement is partitioned as: $hX = h_1 X_1 + h_2 X_2$ and $h^* X^* = h^*_1 X^*_1 + h^*_2 X^*_2$, and if $h^*_1 = h_1$ then the filter/system update corresponding to (14) or (20) becomes

$$\begin{bmatrix} e \\ \hat{X}^*_2 \\ X_2 \end{bmatrix}^+_k = \begin{bmatrix} e \\ \hat{X}^*_2 \\ X_2 \end{bmatrix}_k - \begin{bmatrix} K^*_k \\ --- \\ 0 \end{bmatrix}^- \{[h_1 \vdots h^*_2 \vdots -h_2] \begin{bmatrix} e \\ \hat{X}^*_2 \\ X_2 \end{bmatrix}^-_k - V_k\} \qquad (26)$$

The covariances of (25) and (26) correspond to (15) and (16) with the initial condition, P_o, and the driving noise, Q_k, appropriately partitioned and where:

$$\Phi = \begin{bmatrix} \phi_{11} & \phi^*_{12} & -\phi_{12} \\ 0 & \phi^*_{22} & 0 \\ 0 & 0 & \phi_{22} \end{bmatrix} \quad \text{or with smoothing augmentation} \quad \begin{bmatrix} \phi_{11} & \phi^*_{22} & 0 & -\phi_{12} \\ 0 & \phi^*_{22} & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & \phi_{22} \end{bmatrix} \qquad (27)$$

$H = [h_1 \vdots h^*_2 \vdots -h_2]$ or with smoothing augmentation $[h_1 \vdots h^*_2 \vdots 0 \vdots -h_2]$.

In summary, if part of the dynamics, $\phi_{11} = \phi^*_{11}$, and part of the measurement, $h_1 = h^*_1$ are identical, then only the error state and filter and system states that are different need to be included for fixed-point suboptimal smoothing covariance calculations.

IV. APPLICATION TO NAVIGATION SYSTEM ERROR ESTIMATION

The following example applies fixed-point suboptimal smoothing to design and analysis of a smoother used to survey and estimate the vertical deflection at-sea. The vertical deflection is an uncertainty in the direction of gravity and can be estimated using an inertial navigation system (INS) and accurate external geodetic position fixes. Short term dynamics (time < 4 hrs) of an inertial system can be approximated by [9-10].

$$\dot{\phi} = \frac{\delta V_I}{R} + \epsilon \tag{28}$$

$$\delta \dot{V}_I = -g\,\phi + \delta A_{VD} + \delta A_{VM} \tag{29}$$

$$\delta \dot{P}_I = \delta V_I \tag{30}$$

and the measurement is: $Z_p = \delta P_I - \delta P_R$ (31)

where ϕ = INS Platform Tilt, δV_I = INS Velocity Error, ϵ = Gyro Drift Rate, δA_{VD} = Vertical Deflection Errors, δA_{VM} = Velocity Meter (Accelerometer) Errors, δP_I = INS Position Error, g = Gravity. R = Radius of Earth, δP_R = External Geodetic Position Error.

It is clear from (29) that the vertical deflection and accelerometer errors may not be separable so we will assume that the accelerometers are of better quality than the vertical deflection estimation errors. For this reason the accelerometer errors will be neglected in the analysis. We also will assume that the data taking starts from dockside with a well surveyed dockside vertical deflection (i.e., no uncertainty in the dockside vertical deflection), a well settled INS with no uncertainty in the navigation system initial position, velocity and tilt. The gyros will be aircraft quality and the vertical deflection model for the smoother will be a first order Markov with parameters as shown in Table 1. The values of the parameters in Table 1 for "Smoother" are those values that are used in (1-7) to determine the gains. The actual or system model for the vertical deflection will have the same standard deviation as the model used for the smoother design, but we will analyze the effect of changes in actual vertical deflection correlation time and dimension on smoother performance (see Table 1 for model values).

Table 1. Smoother/System Parameters

Error Source	1σ (P_o matrix)	Correlation Time or Matrix	Driving Noise, q	Comments
Navigation System Initial Conditions				
Position	0.0			Smoother and system
Velocity	0.0			Smoother and system
Tilt	0.0			Smoother and system
Driving Errors				
Gyro Drift Rate	0.01 deg/hr	5 hrs.	0.4×10^{-4} deg^2/hr^3	Smoother and system
Vertical Deflection*				Smoother: T = 2 hrs; q = 144 $\widehat{sec}^2$/hr.
1st Order Markov	0.0	T	$2\sigma^2/T$	System: T varies; q = $\frac{288}{T}$ $\widehat{sec}^2$/hr.
2nd Order Markov	$\begin{bmatrix} 0 & 0 \\ 0 & \frac{\sigma^2}{T^2} \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ -\frac{1}{T^2} & \frac{-2}{T} \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ 0 & \frac{4\sigma^2}{T3} \end{bmatrix}$	Smoother: 1st order Markov above System: T varies; σ = 12 $\widehat{sec}$. (Vertical deflection variance is upper left of 2 x 2 matrix)
Position Measurement				
Rate: ΔT = 0.1 hr.	60 ft. (noise)			Smoother and system

* The vertical deflection is a function of distance but becomes a time function depending on ship's speed [10].

The change in smoothed vertical deflection estimation error covariance for the fixed smoother parameters in Table 1 as a function of varying correlation time of the actual vertical deflection model is shown in Fig. 1. Using the same smoother gains, Fig. 2 shows the effect of varying the "T" parameter in the postulated second-order Markov actual vertical deflection model. The smoothed covariances in Figs. 1 and 2 are calculated at the approximate mid-point of the data interval (at 1.9 hours with a total of 4 hours of data and a measurement interval of 0.1 hours). The gains for the smoother are the optimal gains for the five state model from Eqs. (28-30) and "smoother" conditions in Table 1 until 1.9 hours. At 1.9 hours the state is augmented to six states by the vertical deflection estimation error, and the gains for the six state system are computed until the end of the survey (4 hours).

With the gains for the time period until 1.9 hours, the 4-state error vector, one vertical deflection filter state, and one vertical deflection system state are used to compute the covariance. At 1.9 hours one state is added representing the smoothing error between the smoothed vertical deflection state and the actual vertical deflection state, and the covariance calculations are continued until 4 hours. The covariance of the smoothed vertical deflection estimation error (at 1.9 hours with 4 hours of data) as well as the filtered estimation error (at 1.9 hours with 1.9 hours of data) are shown in Fig. 1 as a function of varying the actual vertical deflection correlation time. Notice that if the actual vertical deflection correlation time is longer than the value used for the smoother design, the smoothed estimation error covariance is less than that predicted by the smoother.

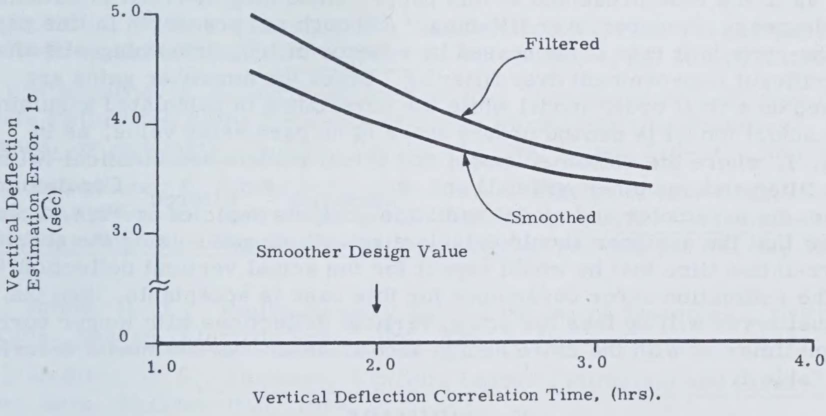


Fig. 1. Filter and Smoother Estimation Error Standard Deviation for Variations in Vertical Deflection Correlation Time

The results shown in Fig. 2 use the same smoother gains as in Fig. 1. For the error analysis of the smoother performance in this case the actual vertical deflection is modeled as a second order Markov with the same variance as the smoother design value while the parameter, T, is allowed to vary (see Table 1 for model structure). Before 1.9 hours, the 4 state error vector, one vertical deflection filter state, and two vertical deflection system states are used. At 1.9 hours, one state is added representing the smoothed vertical deflection estimation error.

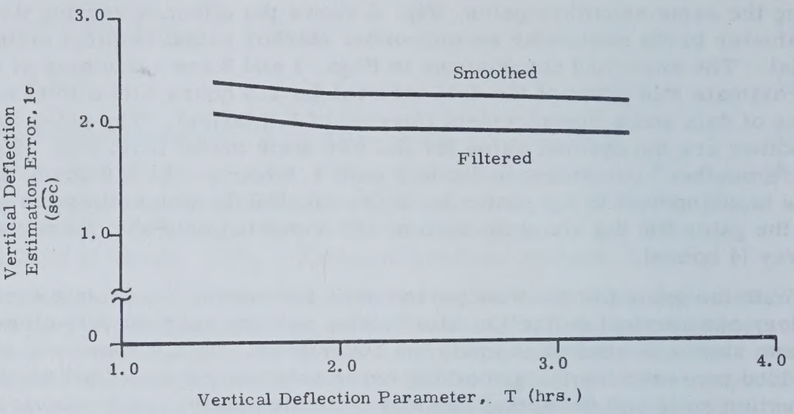


Fig. 2. Filter and Smoother Estimation Error Standard Deviation for Variations in Vertical Deflection Parameter of Second Order Markov

In contrast with Fig. 1 the results in Fig. 2 show that suboptimal smoothing may have a larger variance than suboptimal filtering. This is certainly possible since both the filter and smoother are suboptimal, and if, as in the case presented in this paper, smoothing does not significantly decrease the error over filtering. Although not presented in this paper, if the gyro drift rate is decreased by a factor of ten, smoothing will offer significant improvement over filtering. Since the smoother gains are based on a first order model while the covariance is calculated assuming the actual model is second order, there is no parameter value, as in Fig. 1, where the smoother model and actual models are identical (with the filter and smoother optimal) and $\sigma_{\text{smoother}} \leq \sigma_{\text{filtered}}$. Conclusions from the parameter and model variation analysis depicted in Figs. 1 and 2 show that the designer should calculate smoothing gains using the shortest correlation time that he would expect for the actual vertical deflection. If the estimation error covariance for this case is acceptable, then the actual error will be less for actual vertical deflections with longer correlation times or with the more benign second-order Markov model described in Table 1.

V. SUMMARY

A solution to the suboptimal linear smoothing problem has been derived in Section II based on fixed-point smoothing. The algorithm is relatively uncomplicated and easily implemented since it has the form of the more familiar filter equations. The technique can be used to analyze the effect of mismatch between smoother design values and actual system values for all matrices and parameters used in the design including variations in system structure as well as dimension. Fixed-point suboptimal smoothing is best suited for situations where estimation error covariances are not desired for the entire state vector at every time point.

In many applications part of the smoother dynamics and actual system dynamics have identical structure. For these problems the state size can be reduced by using the error vector described in Section III for smoother analysis.

An example of smoother design and analysis presented in Section IV is the application of at-sea vertical deflection estimation using an aircraft quality inertial navigation system, accurate geodetic position reference, and accurate dockside initial conditions. The results of parameter and model variation trade-offs show that the smoother should be designed for the shortest correlation time expected in the actual vertical deflections. If the smoothed estimation error covariance is acceptable for this case, then the smoothed estimation error covariance will be smaller for longer correlation times and for a second order Markov model as well. The example also shows that suboptimal smoothing can have a larger estimation error covariance than suboptimal filtering. This does not necessarily mean a poor suboptimal smoother design especially if smoothing does not offer significant improvement over filtering.

REFERENCES

[1] Mehra, R. K., "On Optimal and Suboptimal Linear Smoothing", Proceedings of the National Electronics Conference, Vol. XXIV, 1968.

[2] Griffin, R. E. and Sage, A. P., "Sensitivity Analysis of Discrete Filtering and Smoothing Algorithms", AIAA J., Vol. 7, No. 10, pp. 1890-1897, October 1969.

[3] Griffin, R. E. and Sage, A. P., "Sensitivity Analysis of Fixed Point Linear Smoothing Algorithms", Int. J. Control, Vol. 8, No. 4, pp. 321-337, 1968.

[4] D'Appolito, J. A. and Hutchinson, C. E., "Low Sensitivity Filters for State Estimation in the Presence of Large Parameter Uncertainties", IEEE Trans. on Automatic Control, Vol. AC-14, No. 3, pp. 310-312, June 1969.

[5] Hutchinson, C. E. and D'Appolito, J. A., "Design of Low-Sensitivity Kalman Filters for Hybrid Navigation Systems", AGARD Conference Proceedings No. 54 on Hybrid Navigation Systems, AGARD CP No. 54, Jan. 1970.

[6] Rome, H. J., "Finite Memory Batch Processing Smoother", IEEE Trans. Aerosp. Electron Syst., Vol. AES-7, No. 5, September 1971.

[7] Meditch, J. S., Stochastic Optimal Linear Estimation and Control, New York, McGraw-Hill, 1969.

[8] Biswa, K. K., and Mahalanabis, A. K., "An Approach to Fixed-Point Smoothing Problems", IEEE Trans. Aerosp. Electron Syst., Vol. AES-8, No. 5, September 1972.

[9] Britting, K. R., Inertial Navigation Systems Analysis, NY: Wiley, 1971.

[10] Benson, D. O., Jr., "Estimation of High Frequency Vertical Deflection Error", IEEE Position, Location and Navigation Symposium, PLANS '78, San Diego, California, Nov. 6-9, 1978.

ON THE DETERMINATION OF REGRESSION FUNCTIONS

GARY L. WISE
Department of Electrical Engineering
University of Texas at Austin
Austin, Texas 78712

and

NEAL C. GALLAGHER, JR.
School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

ABSTRACT

This paper is concerned with the determination of regression functions from only a partial characterization of the joint distribution. It is shown that statistical information consisting of various moments and joint moments is sufficient to characterize a regression function. An application to regression functionals is also considered.

I. INTRODUCTION

Let X and Y be random variables with Y integrable, i.e. $E\{|Y|\} < \infty$, and consider the regression function of Y on X,

$$m(x) = E\{Y|X=x\}.$$

As is well known, $m(\cdot)$ is a Borel measurable function, and it frequently arises in engineering applications. For example, if Y is a second order random variable, then the minimum mean squared error estimate of Y in terms of X is given by m(X) [1, pp. 77-78].

In some cases $m(\cdot)$ has a particularly simple form. For example, if X and Y are jointly Gaussian with respective means m_X and m_Y, respective variances $\sigma_X^2 > 0$ and σ_Y^2, and correlation coefficient ρ, then

$$m(x) = ax + b, \tag{1}$$

where $a = (\sigma_Y/\sigma_X)\rho$ and $b = m_Y - am_X$. However, in the case of jointly Gaussian random variables, m_X, m_Y, σ_X, σ_Y, and ρ completely determine the bivariate distribution of the two random variables.

In more general cases, the question arises as to how much information about the bivariate distribution is required to determine the regression function. If X and Y are two second order random variables that are separable in the sense of Nuttall [2], then the regression function $m(\cdot)$ has the form given by (1). However, knowing that two second order random variables are separable in the sense of Nuttall, and knowing the means, variances, and the correlation coefficient is not sufficient to determine the bivariate distribution of the two random variables. Notice that any two random variables whose bivariate characteristic function is elliptically symmetric are separable in the sense of Nuttall [3].

As we have seen, there exists a class of joint distributions such that the regression function can be determined knowing that the two random variables belong to that class and also knowing means, variances, and the correlation coefficient. However, it might seem reasonable to conjecture that in more general cases, the regular conditional distribution [4] of Y given X=x is required. Although the regular conditional distribution of Y given X=x is sufficient to determine m(x), in the next section we will see that it is never necessary.

In this paper we will be concerned with statistical information such that there can be only one regression function consistent with the given statistical information. In the next section we consider the regression of Y on a random variable and then on a random vector. Then in the following section we consider the regression functional, that is, the regression of Y on a random process.

II. DEVELOPMENT

Let Y be a second order random variable, let X be a random variable with compact support, and let $m(\cdot)$ be given by Eq. (1). Define the measure μ on the Borel sets of $\mathbb{R}$ by

$$\mu(A) = P(X \in A) ,$$

and let $\|\cdot\|$ denote the $L_2(\mu)$ norm. We will say that a polynomial has max degree N if the degree of the polynomial is no greater than N. We note that for any $\varepsilon > 0$, if N is sufficiently large, there exists a polynomial of max degree N $P_N(x)$ such that

$$\| m - P_N \| < \varepsilon. \tag{2}$$

That is, there exists a continuous function $h(\cdot)$ such that [5]

$$\| m - h \| < \varepsilon/2 ,$$

and by the Weierstrass Theorem there exists a polynomial P_N of max degree N with N sufficiently large such that

$$\| h - P_N \| < \varepsilon/2 .$$

Thus Eq. (2) follows by the triangle inequality. Hence there exists a sequence of polynomials $P_N(x)$ such that

$$P_N(x) \to m(x) \qquad \text{in } L_2(\mu) .$$

Let $Q_N(x)$ be the polynomial of max degree N that is closer to m(x) (in $L_2(\mu)$) than any other polynomial of max degree N. We note in passing that $Q_N(x)$ is uniquely defined a.e. $[\mu]$ by the Projection Theorem. That is, there may exist more than one representation of $Q_N(x)$ (i.e. with different coefficients) but they are all equal a.e. $[\mu]$. From the preceding observations, we have that

$$Q_N(x) \to m(x) \qquad \text{in } L_2[\mu] .$$

Express the polynomial $Q_N(x)$ as

$$Q_N(x) = \sum_{j=0}^{N} a_j(N)\, x^j .$$

It follows from the Projection Theorem that the $a_j(N)$ can be determined from the relation

$$E\left\{\left[m(X) - \sum_{j=0}^{N} a_j(N)\, X^j\right] X^k\right\} = 0, \quad k = 0, 1, 2, \ldots, N .$$

This is equivalent to

$$E\{X^k Y\} = \sum_{j=0}^{N} a_j(N)\, E\{X^{j+k}\}, \quad k = 0, 1, 2, \ldots, N . \tag{3}$$

Thus we have seen that from a knowledge of

$$E\{X^k\}\ , \quad k = 1, 2, \ldots$$

and

$$E\{YX^k\}\ , \quad k = 0, 1, 2, \ \ldots\ ,$$

we can construct a sequence of polynomials $Q_N(x)$ that converge in $L_2(\mu)$ to $m(x)$.

Now let X be an arbitrary random variable. Let g be an invertible Borel measurable function whose range is bounded. Define the random variable $\tilde{X}$ as $\tilde{X} = g(X)$, and the measure $\tilde{\mu}$ on the Borel sets of $\mathbb{R}$ by $\tilde{\mu}(A) = P(\tilde{X} \in A)$. From the above discussion, we see that

$$\tilde{m}(x) = E\{Y | \tilde{X} = x\}$$

is determined a.e.$[\tilde{\mu}]$ by the quantities

$$E\{\tilde{X}^k\}\ , \quad k = 1, 2, \ \ldots \tag{4}$$

and

$$E\{Y\tilde{X}^k\}\ , \quad k = 0, 1, 2, \ \ldots\ . \tag{5}$$

Let $\tilde{Q}_N(x)$ denote the polynomial of max degree N constructed in the preceding fashion. Then

$$\tilde{Q}_N(x) \to \tilde{m}(x) \qquad \text{in } L_2(\tilde{\mu})\ .$$

Notice that $m(x) = \tilde{m}[g(x)]$. From a change of variables result [6, p. 182], we have that

$$\int_{g(\mathbb{R})} [\tilde{Q}_N(x) - \tilde{m}(x)]^2\, \tilde{\mu}(dx) = \int_{\mathbb{R}} [\tilde{Q}_N[g(x)] - m(x)]^2\, \mu(dx)\ .$$

Therefore, $\tilde{Q}_N[g(x)] \to m(x)$ in $L_2(\mu)$.

Now we will remove the restriction that Y be second order. Assume that Y is an integrable random variable and let

$$G_k(y) = \begin{cases} y \text{ if } |y| \le k \\ 0 \text{ if } |y| > k\ . \end{cases}$$

Then $G_k(Y)$ is a second order random variable and [1, p. 23]

$$E\{G_k(Y)|X{=}x\} \to E\{Y|X{=}x\} \quad \text{a.e.}[\mu]\ .$$

Since $|G_k(Y) - Y| \le |Y|$ and $|Y|$ is integrable, we have that $E\{G_k(Y)|X{=}x\} \to m(x)$ in $L_1(\mu)$ by the dominated convergence theorem [6, pp. 124-125].

Thus from a knowledge of the quantities in Eqs. (4) and (5) we can derive a sequence of estimates for $E\{G_k(Y)|X=x\}$ which converges in $L_2(\mu)$, and consequently in $L_1(\mu)$ (see, for example, [7]). Also, $E\{G_k(Y)|X=x\}$ converges to $E\{Y|X=x\}$ in $L_1(\mu)$. Thus, by a straightforward diagonalization procedure, we can derive a sequence of estimates which converges in $L_1(\mu)$ to $m(x)$. These results are summarized in the following theorem.

Theorem 1: Let Y be an integrable random variable, let X be an arbitrary random variable, and let g be an invertible Borel measurable function mapping the reals into a bounded set. Then the regression function m is determined a.e.[μ] by the quantities

$$E\{[g(X)]^k\}\ , \quad k = 1, 2, \ldots$$

and

$$E\{Y[g(X)]^k\}\ , \quad k = 0, 1, 2, \ldots\ .$$

Consider for the moment the case where X and Y are independent. In this case a solution to Eq. (3) is given by

$$a_0(N) = E\{Y\}$$

$$a_j(N) = 0\ , \ j > 0\ ,$$

and we get that $m(x) = E\{Y\}$.

Now consider the following two different bivariate density functions:

$$f_1(x,y) = \frac{1}{\sqrt{2\pi}\,\sigma}\exp\left[-\frac{(y-\rho x)^2}{2\sigma^2}\right] I_{[0,1]}(x)$$

$$f_2(x,y) = xI_{[\rho x-1,\rho x+1]}(y)\ I_{[0,1]}(x),$$

where $\sigma > 0$, $\rho \in (-1,1)$, and I denotes the indicator function. Assuming that the density of (X,Y) is given by f_1, we find that

$$E\{X^k\} = \frac{1}{k+1}$$

$$E\{YX^k\} = \frac{\rho}{k+2}\ .$$

In this case, for $N \geq 1$, a solution to Eq. (3) is given by

$$a_1(N) = \rho \tag{6}$$

$$a_j(N) = 0\ , \ j \neq 1\ , \tag{7}$$

and we conclude that

$$m(x) = \rho x\ . \tag{8}$$

If we assume that the density of (X,Y) is given by f_2, we find that

$$E\{X^k\} = \frac{2}{k+2}$$

$$E\{YX^k\} = \frac{2\rho}{k+3} .$$

In this case, for $N \geq 1$, Eqs. (6) and (7) still satisfy Eq. (3), and the regression function is once again given by Eq. (8). Thus, in this example, the two pairs of marginal densities are not the same, the conditional densities of Y given X=x are not the same, and the moment sequences are not the same; however, the moment sequences are sufficient to characterize the conditional expectations, which are identical. Numerous other similar examples may easily be constructed.

Now we will consider the regression of Y upon a set of random variables. Let X be an arbitrary random vector taking values in $\mathbb{R}^n$, and let μ be defined on the Borel sets of $\mathbb{R}^n$ by

$$\mu(B) = P(X \in B) .$$

Lemma 1: If μ has compact support, then the class of all polynomials is dense in $L_2(\mu)$.

Proof: Let q be an arbitrary element in $L_2(\mu)$. For any $\varepsilon > 0$, there exists [5] a function $h: \mathbb{R}^n \to \mathbb{R}$ which is continuous and has compact support such that

$$\| q-h \| < \varepsilon/2 .$$

By the Stone-Weierstrass Theorem [8] there exists a polynomial p in n variables such that

$$\| h-p \| < \varepsilon/2 ,$$

and thus by the triangle inequality

$$\| p-q \| < \varepsilon .$$

QED

We recall that the degree of a monomial in n variables is the sum of the powers of the variables, and the degree of a polynomial is the degree of the monomial having the largest degree over all the monomials in the polynomial with nonzero coefficients. There are

$$C(n,d) = \binom{n+d-1}{d}$$

monomials of degree d in n variables [9].

Assume that Y is a second order random variable, and define m(x) by Eq. (1), where x is now an element of $\mathbb{R}^n$. Assume that μ has compact support. Let $Q_N(x)$ be the polynomial of max degree N which is closer, in the $L_2(\mu)$ norm, to m(x) than any other polynomial of max degree N.

Consider a monomial in n variables of degree d. There will be C(n,d) of them. Order them lexicographically by the powers of the components of x, and let $m_{jd}(x)$ denote the j-th monomial of degree d. Then $Q_N(x)$ can be expressed as

$$Q_N(x) = \sum_{d=0}^{N} \sum_{j=1}^{C(n,d)} a_{jd}(N) m_{jd}(x) .$$

It follows from the Projection Theorem that the coefficients $a_{jd}(N)$ are given by the solution to the following set of equations:

$$E\{Ym_{ik}(X)\} = \sum_{d=0}^{N} \sum_{j=1}^{C(n,d)} a_{jd}(N)\ E\{m_{jd}(X)\ m_{ik}(X)\}, \tag{9}$$

$k = 0, 1, \ldots, N$ and $i = 1, \ldots, C(n,k)$. If the coefficients $a_{jd}(N)$ satisfy Eq. (9), then it follows from Lemma 1 that

$$Q_N(x) \to m(x) \qquad \text{in } L_2(\mu) .$$

Now we remove the assumption that X has compact support and let X be an arbitrary random vector taking values in $\mathbb{R}^n$. Let g be an invertible Borel measurable function mapping $\mathbb{R}^n$ into a bounded subset of $\mathbb{R}^n$, and let $\tilde{X} = g(X)$. We see that

$$\tilde{m}(x) = E\{Y|\tilde{X}=x\}$$

is determined a.e.$[\tilde{\mu}]$, where $\tilde{\mu}(A) = \mu[g^{-1}(A)]$, by the quantities

$$E\{m_{jd}(\tilde{X})\}$$

and

$$E\{Ym_{jd}(\tilde{X})\}$$

for $d = 0, 1, 2, \ldots$ and $j = 1, \ldots, C(n,d)$. Let $\tilde{Q}_N(x)$ be the polynomial of max degree N determined in the preceding fashion. Then, similar to the development of Theorem 1, we can employ a change of variables result [6, p. 182] to conclude that

$$\tilde{Q}_N[g(x)] \to m(x) \qquad \text{in } L_2(\mu) .$$

A chopping argument as in the development of Theorem 1 allows us to remove the second order restriction on Y. Then a straightforward diagonalization procedure results in a sequence of estimates which converges to m(x) in $L_1(\mu)$. This result is summarized in the following theorem.

Theorem 2: Let Y be an integrable random variable, let X be an arbitrary random vector taking values in $\mathbb{R}^n$, and let g be an invertible Borel measurable function mapping $\mathbb{R}^n$ into a bounded subset of $\mathbb{R}^n$. Then the regression function m is determined a.e.$[\mu]$ by the quantities

$$E\{m_{jd}[g(X)]\} \quad \text{and} \quad E\{Ym_{jd}[g(X)]\}$$

for $d = 0, 1, 2, \ldots$ and $j = 1, \ldots, C(n,d)$.

III. REGRESSION FUNCTIONALS

As before, assume that Y is an integrable random variable, but now let T be an infinite subset of $\mathbb{R}$ and let $\{X(t),\ t \in T\}$ be a random process. Let S denote the space of all extended real valued functions defined on T, and let $\mathscr{B}(S)$ denote the σ-algebra on S generated by the class of all cylinders in S. Let $\mathscr{B}$ denote the Borel sets of $\mathbb{R}$. Then the regression functional

$$m[x(t),\ t \in T] = E\{Y | X(t) = x(t),\ t \in T\}$$

is a measurable function from $(S, \mathscr{B}(S))$ to $(\mathbb{R}, \mathscr{B})$ (see, for example, [10]).

Let μ be the measure induced on $\mathscr{B}(S)$ by $\{X(t),\ t \in T\}$. That is, for any cylinder C in S, $\mu(C) = P(\{X(t),\ t \in T\} \in C)$, and μ is extended to $\mathscr{B}(S)$ via Kolmogorov's Theorem (see, for example, [11]).

It follows from [1, pp. 21, 604] that there exists a countable subset of T, say $\tilde{T} = \{t_1, t_2, \ldots\}$, depending on the random variable Y, such that

$$E\{Y | X(t) = x(t),\ t \in T\} = E\{Y | X(t) = x(t),\ t \in \tilde{T}\} \text{ a.e.}[\mu] .$$

Let

$$M = E\{Y | X(t),\ t \in \tilde{T}\},$$

$$M_n = E\{Y | X(t_1), \ldots, X(t_n)\},$$

$$\mathscr{F} = \sigma\{X(t),\ t \in \tilde{T}\},$$

and

$$\mathscr{F}_n = \sigma\{X(t_1), \ldots, X(t_n)\}.$$

Then from the properties of iterated conditional expectations [1, p. 37], it follows that

$$E\{M_{n+1} | \mathscr{F}_n\} = M_n \quad \text{wp1} ,$$

and hence $\{M_n, \mathscr{F}_n, n \geq 1\}$ is a martingale. It follows from [1, p. 332] that $M_n \to M$ wp1. Since $E\{|M_n|\} \leq E\{|Y|\} < \infty$, it follows from a martingale convergence theorem [1, p. 319] due to Doob that $E\{|M_n - M|\} \to 0$. This is equivalent to

$$E\{Y | X(t_i) = x(t_i),\ i=1,\ldots,n\} \to E\{Y | X(t) = x(t),\ t \in \tilde{T}\}$$

in $L_1(\mu)$. Notice that Theorem 2 is applicable to $E\{Y | X(t_i) = x(t_i),\ i=1,\ldots,n\}$. Thus a straightforward diagonalization procedure results in a sequence of estimates which converges to $m[x(t),\ t \in T]$ in $L_1(\mu)$. This result is summarized in the following theorem.

Theorem 3: Let Y be an integrable random variable and let $\{X(t),\ t \in T\}$ be a random process. Let $\{g_n,\ n=1,2,\ldots\}$ be a sequence of functions where g_n is an invertible Borel measurable function from $\mathbb{R}^n$ to a bounded subset of $\mathbb{R}^n$. Assume that for all positive integers n and for all sets of n points in T, say $t_1, \ldots, t_n$, the quantities

$$E\{m_{jd}(g_n[X(t_1), \ldots, X(t_n)])\}$$

and

$$E\{Ym_{jd}(g_n[X(t_1), \ldots, X(t_n)])\}$$

for $d = 0, 1, 2, \ldots$ and $j = 1, \ldots, C(n,d)$ are known. Then up to μ equivalence, there is only one possible regression functional $m[x(t), t \in T] = E\{Y|X(t) = x(t), t \in T\}$.

ACKNOWLEDGEMENT

This research was supported by the Air Force Office of Scientific Research, Air Force Systems Command, USAF, under Grants AFOSR-76-3062 and AFOSR-78-3605.

REFERENCES

1. J. L. Doob, Stochastic Processes, Wiley, New York, 1953.

2. A. H. Nuttall, "Theory and Application of the Separable Class of Random Processes," Technical Report 343, Research Laboratory of Electronics, Massachusetts Institute of Technology, May 26, 1958.

3. D. K. McGraw and J. F. Wagner, "Elliptically Symmetric Distributions," IEEE Trans. Inform. Th., Vol. IT-14, pp. 110-120, January 1968.

4. L. Breiman, Probability, p. 79, Addison-Wesley, Reading, Mass., 1968.

5. W. Rudin, Real and Complex Analysis, p. 71, McGraw-Hill, New York, 1974.

6. N. Dunford and J. T. Schwartz, Linear Operators Part I: General Theory, Interscience, New York, 1957.

7. M. Loève, Probability Theory, p. 164, Van Nostrand, New York, 1963.

8. J. Dieudonné, Foundations of Modern Analysis, p. 139, Academic Press, New York, 1969.

9. R. W. Brockett, "Lie Algebras and Lie Groups in Control Theory," in Geometric Methods in System Theory, D. Q. Mayne and R. W. Brockett, eds., Reidel, The Netherlands, 1973, pp. 43-82.

10. I. I. Gihman and A. V. Skorohod, The Theory of Stochastic Processes I, p. 34, Springer-Verlag, New York, 1974.

11. P. Billingsley, Probability and Measure, p. 433, Wiley, New York, 1979.

SCHEDULING INDEPENDENT TASKS WITH COMMUNICATIONS*

D. H. CORNETT AND M. A. FRANKLIN
Department of Electrical Engineering
Washington University, Saint Louis, MO 63130

ABSTRACT

The problem of scheduling independent tasks requiring periodic intercommunication is considered. The problem is formulated with a minimum finish time objective function and identical resources. Branch-and-bound algorithms may be used to produce optimal schedules. Since it is unlikely, however, that an optimal solution can be obtained in polynominal time, a fast heuristic has been developed and is presented. The results of an experiment comparing the performance of the heuristic and the optimal are given. In 99.5% of the cases considered, the schedule produced by a particular heuristic had a finish time within 25% of that produced by the optimal algorithm. The worst case difference was 41% and the mean was 2.3%.

I. INTRODUCTION

With the advent of LSI and VLSI technologies for the fabrication of digital logic, there has been an increasing interest in the development of multiple processor computer systems (1). Such systems, with their parallel processing capabilities, have the potential for achieving high computational rates while maintaining low cost.

One of the central questions related to effective use of such systems concerns the design of algorithms appropriate to this new architectural environment. Although this has been investigated for certain special problem classes (2,3), the general question of algorithm design remains.

This paper considers the special class of problems where the solution algorithm can be represented as a set of independent tasks which are periodically required to communicate with each other. Such a problem arises for instance with one method for the parallel solution of ordinary differential equations. This example is examined and the general problem is presented as a task scheduling problem. It differs, however, from previous work (4,5) in that communications overhead are explicitly represented and considered. It is pointed out that the optimal problem is NP-complete (4,5) and thus very likely does not have a polynominal time solution. A heuristic which can handle very large problems is presented. Analytic performance bounds have not yet been obtained; however, the results of an extensive experiment comparing optimal branch-and-bound, and nonoptimal heuristic algorithms indicate that heuristic scheduling is adequate for many situations. These results are discussed and both algorithms are then applied to the differential equations example.

II. PROBLEM DEFINITION

Begin by considering a general task system composed of N independent tasks, T, with task execution times given by T(i), i = 1,2,...,N. An execution time is the time it takes to process the task when it occupies a particular resource. M identical resources are assumed to be present.

While the tasks in this model are independent (i.e. there are no precedence relationships between them) we assume that certain tasks must periodically communicate with each other. In effect, a cyclic processing

---

*This work has been supported in part by National Science Foundation Grant MCS 78-20731.

environment exists which consists of two phases. During the first phase, task execution, the tasks are processed on the available resources in the manner determined by the task/resource allocation. During the second phase, communication, the tasks exchange information. The model assumes that the order of communication is unimportant and that execution time, Tp, and communication time, Tc, remain constant from cycle to cycle. That is, once a task/resource allocation has been performed, it is not changed until the requisite number of cycles has been completed. The number of cycles required is dependent on the particular iterative algorithm involved and may involve for instance some convergence criterion.

To specify which tasks require intercommunications, define an N·N matrix, A, referred to as the task communications matrix, such that:

$$A = [a(i,j)] \text{ where } a(i,j) = \begin{cases} 1, & \text{if task } i \text{ communicates to task } j,\ i \neq j. \\ 0, & \text{otherwise.} \end{cases}$$

When two tasks i and j require communications (i.e. a(i, j) = 1) and the tasks have been assigned to two different resources for processing, then associated with this communications is a time Tct. For simplicity this time is assumed to be a constant which is the same for all task pairs requiring communications.

Define next an N·M task allocation matrix, X, such that:

$$X = [x(i,j)] \text{ where } x(i,j) = \begin{cases} 1, & \text{if task } i \text{ is assigned to resource } j \\ 0, & \text{otherwise} \end{cases}$$

and

$$\sum_{j=1}^{M} x(i,j) = 1 \qquad \text{, all } i$$

This latter constraint insures that a task will be allocated to a single computer.

The task allocation problem is to determine the matrix X such that some objective function is minimized or maximized. The objective function of interest is the problem solution time, where a problem is considered solved after it has gone through L execution-communication cycles. If Ci is the time for a single cycle i, and Tq is the time for q cycles then the objective function to be minimized is:

$$Tq = \sum_{i=1}^{q} Ci = q\cdot Ci = q\cdot(Tp + Tc)$$

Since q is an integer independent of the particular schedule, the objective function to be minimized is simply the sum of Tp + Tc.

The execution or processing time, Tp, is defined as:

$$Tp = \max_{j=1,2,\ldots,M} \left\{ \sum_{i=1}^{N} T(i)\cdot x(i,j) \right\}$$

This is the standard Minimum Finish Time (MFT) objective function (6).

Note that the way the model has been formulated, communications cannot occur until all task executions have been completed. Whether or not this is realistic depends on the resource intercommunications network available. In any case, this assumption appears to simplify the problem somewhat and allows it to be considered as an extension to the independent task MFT problem.

For the single bus situation Tc can be derived from the previous definitions:

$$Tc = Tct \cdot \sum_{i=1}^{N} \sum_{k=1}^{N} a(i,k) - Tct \cdot \sum_{i=1}^{N} \sum_{k=1}^{N} a(i,k) \cdot \sum_{j=1}^{M} x(i,j) \cdot x(k,j)$$

The first term represents the total communications time requirement if all task pairs requiring communications had the tasks of each pair on different resources. The second term represents the time saved when task pairs reside on the same resource.

The scheduling problem is therefore to find the assignment of tasks to resources (i.e. find X) which minimizes Tp + Tc. Two approaches are considered. The optimal approach requires solving a large combinatoric problem. Unfortunately, for problems of reasonable size this is too costly, and it is likely that the optimal problem does not possess polynomial time solutions (4,5). The second approach requires developing and testing a fast scheduling heuristic. Such a heuristic is discussed and the results of testing presented later in this paper. What follows is an example which illustrates the execution/communication model developed above.

III. AN EXAMPLE

As an example of a problem and solution method which yields a set of independent tasks with periodic communications requirements consider the following set of coupled, first order differential equations.

$$\dot{X1} = f1(X6, t)$$

$$\dot{X2} = f2(X3, X5, X6, t)$$

$$\dot{X3} = f3(X2, X4, X5, t)$$

$$\dot{X4} = f4(X3, X5, t)$$

$$\dot{X5} = f5(X2, X3, X4, t)$$

$$\dot{X6} = f6(X1, X2, t)$$

$\dot{Xi}$, $i \in (i, \ldots, 6)$ represents the derivative of the independent variable with respect to t, while $fi(Xj, Xk \ldots, t)$, $i,j,k \in (1, \ldots, 6)$ is a general function of the independent variables Xi and the dependent variable t. The function evaluations correspond to the task execution times T(i). Since such function evaluations are often the dominant time factor in solving sets of differential equations this is usually a reasonable assumption. Coupling between the equations corresponds to specification of the intertask communications matrix A, as given below.

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Thus task 3 must communicate with tasks 2,4 and 5 (i.e. function f3 requires X2, X4 and X5 as inputs).

Each equation can be thought to be allocated to a resource which performs the integration and function evaluation operations. A cycle corresponds to an integration interval, and the communications corresponds to transfering the appropriate X values between the processors at the completion of each cycle. Typically, for standard initial value problems, cycles will continue until some final time is reached. Thus, in a general way, the independent task allocation problem with communications can be considered to be a model for examining the question of solution of sets of ordinary differential equations on multiple processor systems (7).

Considering the example somewhat further, associate the execution times of Table 1 with the six tasks.

	T(1)	T(2)	T(3)	T(4)	T(5)	T(6)
Task Execution Time	800	600	500	400	150	80

Table 1: Example Task Execution Times

Assume that four identical processors are available (i.e. M=4), that they communicate via a single bus, and that Tct is 160 time units for each variable transferred. Figure 1 presents some possible task schedules and their resulting cycle times.

In the first case, Figure 1a, all tasks are allocated to a single processor. This results in zero communications time and the overall time T is simply the sum of the execution times, T=2530. The second schedule, Figure 1b, uses four computers. The processing time, Tp, has been reduced to 900 units by overlapping execution of various tasks on the separate processors. However, this allocation has resulted in a communication time of 1600 units, which is much larger than the execution time. The overall time, 2500 time units, is still slightly lower than the single processor case. Figure 1c shows how the shift of a single task, the short task 6, from processor 4 to processor 2 greatly improves the cycle time by reducing the communication time.

IV. SCHEDULING ALGORITHMS

One way of finding an optimal schedule is to perform an ordered search of combinations of tasks on the available processors. Branch-and-bound type of algorithms have been used for this purpose (6,7,8). One particular branch-and-bound algorithm which has been implemented with this problem in mind is discussed in reference 9. Algorithm performance, and improvements due to better initial schedules are considered in this reference. Unfortunately, even with tree pruning and with improvements due to good initial schedules, there is an explosion in the number of combinations which must be examined when problems of reasonable size are considered.

This is due to the fact that the scheduling problem presented is NP-complete and thus very likely does not possess polynomial-time solutions. Therefore, it is probably more fruitful to examine heuristic solution methods; one such method is presented in the next section.

To handle a reasonable number of tasks and computers it is necessary to develop scheduling heuristics. This section presents a heuristic which is an extension of the well known Longest Processing Time (LPT) first heuristic (10,11). The heuristic is composed of the following steps.

Step 1: Associate with each task an Augmented Task Execution Time Ta(i):

Ta(i) = f(T(i), Tct, A)

Ta(i) is a function of T(i), Tct and the matrix A, and expresses the impact of task i on the task assignment process. One simple way of doing this is as follows:

$$Ta(i) = K1 \cdot T(i) + K2 \cdot Tct \cdot \sum_{j=1}^{N} (a(i,j)+a(j,i))$$

where K1 and K2 are constants. The vector of augmented task execution times is given by Ta.

$$Ta = [Ta(i)], i = 1,2, \ldots, N$$

Step 2.0: Order Ta from left to right in descending order.

If there are ties in Ta, select the task with minimum T(i)/Ta(i) first. The effect of this will be to place a higher value on those tasks with maximum communications impact. Deal with ties in T(i)/Ta(i) by selecting the task with the maximum number of communications to previously-ordered tasks first (i.e., the task i with maximum $\sum_{j} a(i,j) + a(j,i)$, $j \in$ (Set of previously-ordered tasks)). The emphasis is on giving priority to tasks which communicate heavily with previously ordered tasks. By considering heavily linked sets of tasks as a group, the algorithm tends to assign the group to a single resource. If a tie occurs at this level make an arbitrary ordering choice among the tied tasks.

Step 3.0: Starting with the left most task in Ta and proceeding in a left to right manner allocate the tasks to the resources as follows:

3.1: If the task considered, task i, doesn't communicate with any previously allocated job, use LPT scheduling to schedule the task on a computer.

3.2: If the task i has any communication with any previously allocated job try placing the task on each computer in turn and allocate the task to the resource which minimizes the maximum finish time (MFT) for those tasks, including i, which have thus far been assigned.

If there is a tie in MFT, then on each resource involved in the tie, count the number of tasks that task i communicates with. Place task i on the resource having the highest communications count. If a tie still remains make an arbitrary choice between the tied resources.

Figure 2 shows optimal and heuristic schedules for the example previously presented. The following section discusses the performance of this heuristic.

V. PERFORMANCE EVALUATION

There are two aspects of heuristic performance which must be considered. The first concerns how the heuristic performs the scheduling task with respect to the objective function when compared to the optimal algorithm. The second concerns the computational complexity of the heuristic when compared to the optimal.

Statistical Comparison

With regard to relative scheduling performance it would be advantageous to have an analytic nonstatistical performance bound. Such bounds have been achieved in the simpler case of independent task allocation without communications (10,11,12). Unfortunately deriving such performance bounds appears to be extremely difficult when complexities are added to the basic independent task model and such a bound has not been found for the model presented in this paper. A statistical study has, however, been done and an experimentally derived bound has been obtained. The experiments and the resulting bounds are given below.

The experiment consisted initially of randomly generating a set of configurations. A given configuration is defined by the following parameters:

1. N - The number of tasks: $N \in (2,9)$
2. M - The number of processors: $M \in (2,N)$
3. $T(i)$, $i = 1,2, \ldots, N$ - Task execution times; $50 \leq T(i) \leq 1000$
4. A-matrix - Task communications matrix:

$$a(i,j) = \begin{cases} 1 \text{ with probability } p \\ 0 \text{ with probability } 1-p \end{cases}$$

For each randomly generated N,M and T(i), four different A-matrices were randomly generated with p = .25,.50,.75 and 1.0 respectively. Associated with each of these configurations is also a set of about twenty five different communications times, Tct. These times were equally spaced from 0 to a maximum value, Tct(max). This maximum value was taken to be the value beyond which the optimal allocation resulted in no interprocessor communications taking place. That is, it is that communication time which forces all communicating tasks to be allocated to the same computer. It is not yet known how to get Tct(max) prior to actually obtaining the optimal schedules with different communication times. An empirically derived approximation however, which yielded Tct(max) to within about 10% of the true value was used to determine the incremental values of Tct to be used, with the final range of Tct being determined upon execution of the optimal algorithm. The approximation is given below.

$$Tct(max) \simeq \left[\frac{\sum_{i=1}^{N} T(i)}{N} \right] \cdot \max[(1-D), 0.1]$$

D is a measure of the density of the task communications matrix and is given by:

$$D = \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} a(i,j)}{N(N-1)}$$

Thus for each N, M and T(i), about one hundred variations relating to the A matrix and communications time were specified. Each one of these variations represents a sample used in the experiment. For each sample an optimal schedule and two heuristic schedules were computed using the algorithms presented in the previous sections. The two heuristics differed in the value associated with the constant K2, the first heuristic using K2 = 0 and the second K2 = 1. Both had K1 = 1. The finish times, T, were then compared, tabulated and graphed. The performance measure of interest is the error of difference between the final times as obtained from the heuristics and the optimal. These are defined as follows:

$$E1 = \frac{T(\text{heuristic, } K1 = 1,\ K2 = 0) - T(\text{optimal})}{T(\text{optimal})}$$

$$E2 = \frac{T(\text{heuristic, } K2 = 1,\ K2 = 1) - T(\text{optimal})}{T(\text{optimal})}$$

$$E3 = \min(E1, E2)$$

Table 2 shows the results of this comparison for 821 configurations entailing 20,509 samples. Based on the number of configurations used, standardized differences of means greater than 0.015 are detectable with a 99% confidence.

	Percent of sample with error $\leq$ x (the table entry)				
Error	90%	95%	99.5%	Worst Case	Mean Error
E1	.11	.16	.29	.41	.0336
E2	.11	.16	.32	.75	.0349
E3	.08	.12	.24	.41	.0228

Table 2: Heuristic/Optimal Performance Comparison

Figure 3 indicates the fraction of samples which have a given error. For example, for 48% of the samples the heuristic schedule was as good as the optimal. While there is no significant difference in mean error performance of the heuristic with the two K2 values selected, there is a significant improvement when the minimum of the two heuristic schedules is used. For this case, 99.5% of the samples were within 25% of the optimal when the best of the two heuristics was used. In addition the maximum error in this case was .41 while the mean error was .0228.

Figure 4 and Table 3 indicate how the E3 error varies with the number of tasks $(N \in (5,9)0$.

	Percent of sample with error $\leq$ x (the table entry)				
N	90%	95%	99.5%	Worst Case	Mean Error
5	0.06	0.09	0.21	.29	.015
6	0.07	0.09	0.18	.27	.017
7	0.10	0.15	0.28	.40	.031
8	0.10	0.14	0.24	.41	.034
9	0.10	0.15	0.30	.33	.040

Table 3: E3 Error vs. Number of Tasks

In this case there is a weak relationship between heuristic performance and number of tasks, with the performance degrading as the number of tasks increases. This is somewhat surprising since one would generally expect better heuristic performance as the number of tasks grows. Coffman and Sethi (11) have shown that this is true for straight LPT schedules in a standard independent task, no communications environment. It appears, however, that communications considerations present enough added complexity that the heuristic presented is not able to take advantage of greater scheduling flexibility which more tasks allows.

Table 4 presents the data broken down in terms of E3 error versus the normalized communications time. This normalized communications time is defined as the communications time, Tct, divided by Tct(max). The data here indicate that the heuristic does increasingly well when the communications time ie either very small or very large. When the communications time is small the heuristic is performing the LPT scheduling algorithm and the results fall within both the predicted bounds and actual LPT scheduling results. When the communications time is large the heuristic will tend to allocate tasks to the same resource to minimize intercommunications time. That is, the heuristic is good at deciding on a single resource allocation when that is the optimal. It is when the communication times are moderate that it is most difficult to decide on the best allocation, and it is here that the heuristic does most poorly.

The method of forming the augmented task vector, Ta, and the manner of deciding ties during the scheduling process can have a significant effect on the heuristic's performance. Since finding a heuristic schedule is so much faster than finding an optimal schedule, it is worth considering the performance gain which can result from using the best of several heuristic variations. Indications are (9) that by starting with a few significantly different rankings which arise from selection of differing K1, K2 pairs, the minimum of two or three heuristics can lead to an improvement in overall performance. For instance, using the three rankings (.5,1), (1,-.5) and (1,-1) the mean error is cut in half.

Percent of sample with error $\leq$ x (the table entry)

Normalized Communication Time	90%	95%	99.5%	Worst Case	Mean Error
= 0.0	0.02	0.04	0.09	.08	.005
0.0-0.1	0.04	0.06	0.11	.14	.010
0.1-0.2	0.05	0.08	0.17	.31	.015
0.2-0.3	0.08	0.11	0.20	.34	.022
0.3-0.4	0.09	0.13	0.28	.39	.028
0.4-0.5	0.11	0.16	0.30	.40	.034
0.5-0.6	0.13	0.17	0.28	.41	.037
0.6-0.7	0.12	0.16	0.24	.33	.035
0.7-0.8	0.10	0.13	0.19	.24	.031
0.8-0.9	0.07	0.09	0.13	.21	.021
0.9-1.0	0.07	0.08	0.09	.22	.009
= 1.0	0.00	0.04	0.15	.26	.003

Table 4: E3 Error vs. Normalized Communication Time

Computational Complexity

With regard to computational complexity and speed of execution, the heuristic is dramatically better than the optimal algorithm, especially as the size of the scheduling problem grows. The approximate number of operations needed to perform the heuristic allocation, N(H), and the approximate maximum number needed to perform the optimal allocation, N(O), are given in the equations below (9):

$$N(H) \propto N^2 + N \cdot M$$

$$N(O) \propto M^N \cdot (N^2 + 6N + M)$$

N and M above correspond to the number of tasks and computers respectively. N(O) is an approximation for the maximum number of allocations which must be tried for the optimal branch-and-bound algorithm given in reference 9. While the tree pruning properties of the optimal algorithm may reduce N(O) somewhat (e.g. factor of 2 to 4), the heuristic is still about an order of magnitude faster for problems of complexity (N,M) = (4,4), and is at least three orders of magnitude faster for problems of complexity (N,M) = (6,5) or (7,3). This relative speed advantage of the heuristic grows even faster as larger N and M increase further.

VI. SUMMARY AND CONCLUSIONS

This paper has considered the problem of scheduling independent tasks where there are communications requirements between the tasks and where a minimum finish time objective function is used. The number of combinations which must be examined for a branch-and-bound optimal algorithm grows exponentially, and therefore such algorithms are limited to relatively small problems. This can be a serious drawback in the sample application which deals with the scheduling of independent program tasks in a multiple resource system.

To overcome this problem a fast heuristic algorithm was developed. Experiments comparing heuristic and optimal allocations indicate that for 99.5% of the sample systems tried the heuristic finish times were within 24% of the optimal ones. Furthermore the worst case heuristic was within 41%, and the mean was within 2.3% of the optimal. Picking the best of several fast heuristic executions each having different K1 and K2 values can lead to a significant improvement of the results cited above. When compared with an optimal branch-and-bound algorithm, one incurs a relatively small execution time penalty when running the heuristic several times. This is because the heuristic is at least several orders of magnitude faster than the optimal algorithm for problems where, for instance, six tasks and five resources are present.

REFERENCES

1. Poole, W.G. Jr. & Voigt, R.G."Bibliography 35-Numerical Algorithms For Parallel & Vector Computers"Computing Reviews 15,10(OCT 1974) 379-388.
2. Miranker, W.L. "A Survey of Parallelism in Numerical Analysis" SIAM REV. 13 (1971) 524-547.
3. Trilling, T. "An Architecture for Parallel Processing of Sparse Data Streams" Proc. Int. Conf. Para. Proc. (1979) 298-305.
4. Karp, R.M. "Reducibility Among Combinatorial Problems" in Complexity of Computer Computations, Plenum Press, New York (1972).
5. Sahni, S.K. "Algorithms for Scheduling Independent Tasks" JACM 23, 1 (JAN 1976) 116-127.
6. Coffman, E.G. Jr. (Ed.) Computer and Job/Shop Scheduling Theory, John Wiley & Sons, New York (1975).
7. Franklin, M.A. "Parallel Solution of Ordinary Differential Equations" IEEE Trans. on E.C., C-27,5 (May 1978) 413-420.
8. Kohler, W.H. and Steiglitz, K. "Characterization and Theoretical Comparison of Branch-and-Bound Algorithms for Permutation Problems" JACM 21,1 (1974) 140-156.
9. Cornett, D.H. "Multi-Processor Scheduling of Independent, Communcation Tasks" M.S. Thesis, Dept. of Electrical Engineering, Washington University, St. Louis (May 1977).
10. Ullman, J.D. "NP-Complete Scheduling Problems" J. Computer and System Sciences 10, 3 (JUNE 1975) 384-393.
11. Graham, R.L. "Bounds on Multiprocessing Timing Anomalies" SIAM J. Appl. Math. 17,2 (MAR 1969) 416-429.
12. Coffman, E.G. Jr. and Sethi, R. "A Generalized Bound on LPT Sequencing" Proc. Inter. Symp. on Comp. Performance Modeling, Measurement and Evaluation, Harvard Univ. (MAR 1976) 306-310.

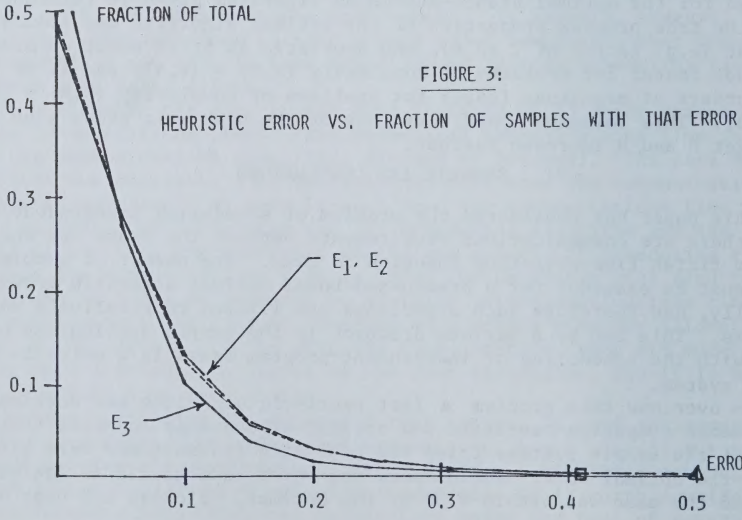
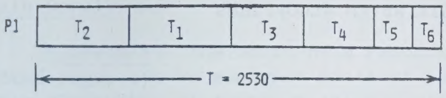
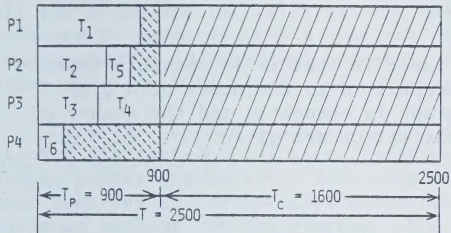


FIGURE 3:

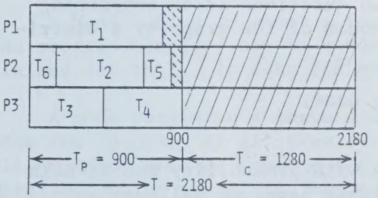
HEURISTIC ERROR VS. FRACTION OF SAMPLES WITH THAT ERROR


P1
T2
T1
T3
T4
T5
T6
T = 2530

A. ALL TASKS ON A SINGLE PROCESSOR

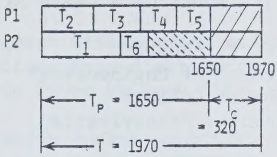

P1
T1
P2
T2
T5
P3
T3
T4
P4
T6
900
2500
Tp = 900
Tc = 1600
T = 2500

B. AN ALLOCATION ON FOUR PROCESSORS

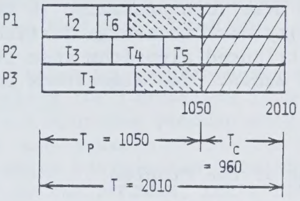

P1
T1
P2
T6
T2
T5
P3
T3
T4
900
2180
Tp = 900
Tc = 1280
T = 2180

C. AN ALLOCATION ON THREE PROCESSORS

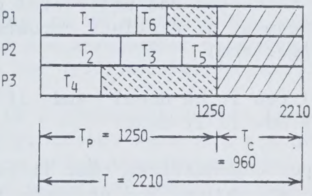
FIGURE 1. AN EXAMPLE


P1
T2
T3
T4
T5
P2
T1
T6
1650
1970
Tp = 1650
Tc = 320
T = 1970

A. AN OPTIMAL SCHEDULE


P1
T2
T6
P2
T3
T4
T5
P3
T1
1050
2010
Tp = 1050
Tc = 960
T = 2010

B. A HEURISTIC SCHEDULE ($K_1=1, K_2=1$)


P1
T1
T6
P2
T2
T3
T5
P3
T4
1250
2210
Tp = 1250
Tc = 960
T = 2210

C. A HEURISTIC SCHEDULE ($K_1=1, K_2=0$)

FIGURE 2: OPTIMAL AND HEURISTIC SCHEDULES FOR THE EXAMPLE

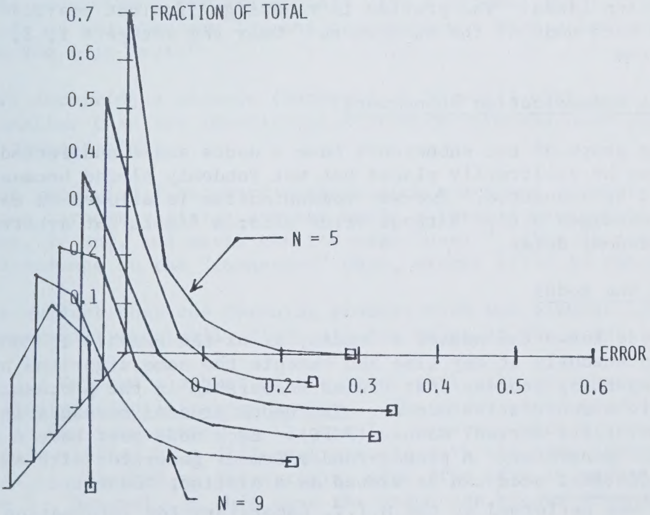


FIGURE 4: HEURISTIC E_3 ERROR VS. FRACTION OF SAMPLES WITH THAT ERROR, PARAMETERIZED BY NUMBER OF TASKS

NUMBERING THE NODES OF A GRAPH BY DISTRIBUTED ALGORITHMS

MAURICE F. ABURDENE
Department of Engineering*
Swarthmore College
Swarthmore, Pennsylvania

ABSTRACT

The main idea presented in this paper is that it is possible (with probability one) to assign distinct identities to the nodes of a graph by a distributed algorithm when the graph is connected and the number of nodes is known. Two algorithms to that effect are given.

Statement of the Problem

Let G be a graph with n nodes and e edges. By the term graph is meant a connected and undirected graph. Two questions arise concerning the assignment of distinct numbers to the nodes of the graph by a distributed algorithm:

(1) Can it be done? And, if it can be done,
(2) How is it done?

This paper demonstrates that it can be done with probability one if n is known to all nodes, and presents two algorithms based on Gallager's distributed algorithms for finding a leader in a network [1,2].

An Application for the Problem

Think of the graph G as a communication subnetwork with n nodes and e communication links. The problem is to assign distinct positive integers to each node of the subnetwork. Only the integers 1, 2, ..., n should be used.

Model of the Communication Subnetwork

Let the graph of the subnetwork have n nodes and e undirected edges. The edges can be arbitrarily placed but not randomly placed because the graph should be connected. Two-way communication is allowed on each edge. The messages arrive without error after a finite but arbitrary (possibly random) delay.

Behavior of the Nodes

Each node knows the number of nodes, n, of the graph. The nodes can wake up spontaneously at any time and execute the same algorithm by starting a spanning process that claims membership in the process. The nodes work in a cooperative manner. The nodes process messages in a first-come, first-served, manner (FCFS). Each node must have a true random number generator. A pseudo-random number generator with an associated distinct seed can be viewed as a distinct identity.

---

* This work was performed at the M.I.T. Laboratory for Information and Decision Systems, Cambridge, Massachusetts.

Main Result

Theorem 1: The nodes of a connected graph with n nodes and e undirected edges can be assigned distinct identities by a distributed algorithm when the number of nodes, n, of the graph is known by each node.

Proof: Assume each node has a true random number generator. Assume all the nodes are initially asleep. Nodes wake up spontaneously or are awakened by messages from a spanning process. Nodes that are awake cannot start a new spanning process. The nodes execute the requests of other nodes in FCFS order. All nodes execute the same algorithm.

Case 1: Suppose a subset of the nodes wakes up spontaneously and chooses distinct initial identification numbers for themselves from the random number generators. Each node starts a spanning process with the same identity as the identification number and travels around the graph claiming members for the process. The messages contain the identity of the process, the number of members claimed by the process, and a bit indicating whether the message is forward or backward.

A node waking up sends a forward message on an unused edge, with the process identity set to a random number and the number of nodes claimed set to 1. It sets the edge status to "used" and labels itself "1".

A node receiving a forward message with a process identity larger than the identity of all previous visiting processes sets the status of all edges to "unused," except the one edge on which the message was received, which is set to "connected." The number of nodes claimed by the process is updated and used as the node identity. A forward message is sent on an unused edge, if any, which is then marked "used." If there is no unused edge, a backward message is sent on the "connected" edge.

A node receiving a forward message with a process identity equal to the identity of the previous largest process sends back a backward message and markes the edge "out."

A node receiving a message (backward or forward) with a process identity smaller than the identity of a previous process does nothing; the lower process dies.

A node receiving a backward message with a process identity equal to the previous largest visiting process sends a forward message on an unused edge, if any, and marks the the edge "used." Else it sends a backward message on the "connected" edge, except if it is the root.

It is evident that the spanning process with the highest identification number will eventually claim all the nodes of the graph.

Case 2: Suppose a subset of the nodes that wake up choose non-distinct initial identification numbers. The nodes start spanning processes in the same manner as in Case 1. However, when a process visits a node visited by a different process with the same identity, the process is returned on the arriving edge (the edge is marked "out"), as in Case 1. However, in this case the graph can become disconnected.

See Figures 1(a) and 2(a). If the graph becomes disconnected, there will be more than one process traversing the subgraphs. Each process will eventually visit every node in the subgraph by traversing all the edges in the subgraph. At this point the root node determines that the number of nodes claimed by the process is not equal to n and the node chooses a new identification number and starts a new spanning process. This procedure continues until one spanning process visits all the nodes. Since each node has a true random number generator, one of the surviving processes will claim all the nodes with probability one. An algorithm based on the above proof is difficult to analyze but clearly contains O(ne) messages.

Algorithm 1: A distributed algorithm with O(ne) messages.

This algorithm is a variation of a distributed processing algorithm for choosing a leader in a network by Gallager [1]. A sequence of steps of the algorithm explain it best.

Case 1: An asleep node wakes up spontaneously. The node performs the following operations:

(a) It chooses an identification number, m, by using its random number generator,
(b) it numbers itself "1", and
(c) it sends a message on an unused edge to another node.

Case 2: The receiving node was asleep. The node performs the following operations:

(a) It wakes up,
(b) it assigns itself the identity of the sending node plus one,
(c) it assigns the spanning process the identity m, and sends a message on an unused edge.

Case 3: The receiving node was awake and visited by a process with a higher identity. The node does nothing, effectively killing the visiting process.

Case 4: The receiving node was awake and visited by a process with a lower identity. The node performs the following operations:

(a) It updates the spanning process identity to the visiting process identity, and
(b) it assigns itself a new identification number equal to the sending process plus one.

Case 5: The receiving node was awake and visited by a process with the same identify. The node performs the following operations:

(a) It marks the edge "out," and
(b) it sends the message back to the sending node for sending on an unused edge.

When the receiving node completes its processing but has no unused edges, it sends the message back on the receiving edge. Also the nodes check to see if the length of the process is n. If it is, then the algorithm terminates.

Most of the cases are self-explanatory except for Case 5. The reason we mark an edge "out" when a node is visited by a process with the same identification number is because it is difficult to distinguish between cases, as shown in Figure 1. The results of Case 5 are shown in Figure 2.

It should be pointed out that cases such as in Figure 1(a) disconnect the network, and the algorithm assigns identification numbers to the subgraphs. When a process spans its subgraph and determines that the length is less than n, it chooses a new identity and starts a new spanning process. However, cases such as in Figure 1(b) do not disconnect the graph, and the algorithm terminates with each node choosing at most one random number.

The worst case, communication cost, is difficult to analyze, but is O(ne) messages.

Algorithm 2: A distributed algorithm with O(e) + n log n messages.

A slight modification to the Gallager algorithm [2] can be used to assign distinct numbers to the nodes of the graph in a distributed way with O(e) + Q(n log n) messages. A brief description of a modified Gallager algorithm follows.

Assume all nodes are initially asleep, but arbitrary subsets wake up and assign themselves identification numbers, m's, from a random number generator and start the asynchronous distributed algorithm. Each node that wakes up spontaneously starts a process, P_m, which traverses the network, and claims and numbers the nodes it traverses. Sleeping nodes are awakened by the receipt of messages and are owned by the process that awakened them. As in the Gallager algorithm, if two processes with distinct identities meet, the process that has assigned a smaller number of nodes "dies." Sometimes this requires tracing the processes and waiting. If both processes have the same number of nodes, the one with the smaller identity dies. The variation to the Gallager algorithm occurs when a process P_m visits a node that is owned by P_m. It is rejected along the edge it arrived on and no attempt is made to determine if the node was visited by the same process P_m or another process with the same identity m.

If the node was visited by the same process, the result is the same as the Gallager algorithm and the network remains connected. However, if the node is visited by a different process but with the same identification number, the effect is to disconnect the edge it arrived on and possibly partition the network into subnetworks and continue the Gallager algorithm. The Gallager algorithm terminates for each subnetwork. A modification to the terminating criterion is included by also checking if the length of the process P_m is n. If it is, the algorithm terminates. If the length of the process P_m is not equal to n, the node chooses a new identification number and reclaims the same nodes by assigning them new identification until it reaches an "out" edge and sends a message along that edge and continues the Gallager algorithm.

The reason that an edge is marked "out" when a node is visited by a process with the same identification number is again because it is difficult to distinguish between cases such as in Figure 1. It has been shown by Gallager that giving preference to the process that has claimed the most nodes, instead of to the process with the largest identity (as in Algorithm 1), reduces the worst-case total number of messages to O(e) + O(n log n), when the network does not become disconnected. If the network becomes disconnected, the proof is still the same and is of O(e) + O(n log n), because processes with the same identification number were forbidden to retrace each other.

It should be pointed out that both algorithms are dependent on the random number generator characteristics and there is a possibility for the algorithm not to terminate when two or more processes keep choosing the same identification number. Also, both algorithms implicitly result in a spanning tree having been created. In a data communication network, this could be used to exchange control information efficiently.

Algorithm 2 will usually require more communication time but fewer communication messages than Algorithm 1.

Summary

It has been shown that it is possible to number the nodes of the graph with n distinct integers by a distributed algorithm with probability one. Two algorithms were described. The algorithms are modifications of Gallager's algorithms.

A related but unsolved problem is: Can we assign distinct numbers to a connected graph by a distributed algorithm when the number of nodes, n, is unknown by each node?

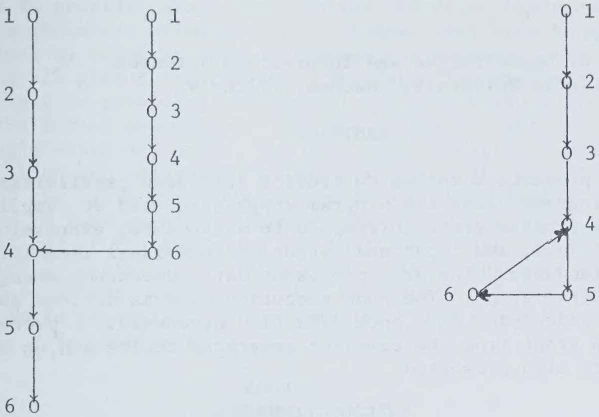
References

[1] Robert G. Gallager, "Choosing a Leader in a Network." Internal Memorandum, LIDS, M.I.T., Cambridge, MA 02139.

[2] Robert G. Gallager, "Finding a Leader in a Network with O(e) + O(n log n) Messages." Internal Memorandum, LIDS, M.I.T., Cambridge, MA 02139.

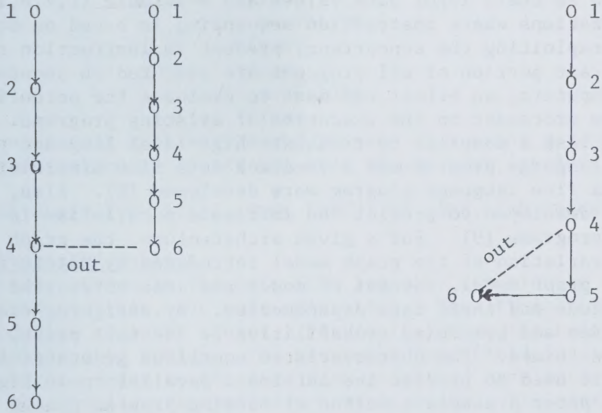
Acknowledgments

The author would like to acknowledge the helpful discussions with Professors Pierre Humblet and Robert G. Gallager.


1
2
3
4
5
6
1
2
3
4
5
6
1
2
3
4
5
6

(a) (b)

Figure 1


1
2
3
4
out
5
6
1
2
3
4
5
6
1
2
3
4
out
5
6

(a) (b)

Figure 2

COMPILE TIME PREDICTION OF INTRINSIC PARALLELISM IN HIGH LEVEL PROGRAMS

C. RETNADHAS
Department of Quantitative and Information Sciences
Western Illinois University, Macomb, Illinois

ABSTRACT

This paper presents a method to predict intrinsic parallelism present in high level programs using the program graph generated at compile time. The nodes in the program graph correspond to statements, expressions, procedure invocations, and input and output of individual identifiers in the high level language. The arcs represent data dependency among the nodes in the program graph. The node execution time is derived assuming execution on an underlying feed back data flow processor. A procedure to form the program graph from the compiler generated tables and an example program graph are also presented.

INTRODUCTION

The sequential control flow sequencing of instructions in a von Neumann type computer limits the amount of concurrency present in a program structure to a minimum. An alternative method to the sequential control flow computer organization is one in which instructions are executed as soon as their input data values are available (1,4,8,10). Computer organizations where instruction sequencing is based on data flow are capable of exploiting the concurrency present in instruction streams. Since a significant portion of all programs are executed on sequential control flow computers, an effort was made to evaluate the potential impact of the data flow processor on the execution of existing programs. To accomplish this task a compiler to translate high level language programs to a data flow language program and a feedback data flow simulator to execute the data flow language program were developed (8). Also, a program graph model was developed to predict the intrinsic parallelism in high level language programs (9). For a given architecture, the graph model is formed using a variation of the graph model introduced by Matrin and Estrin (6,7). In this graph model, the set of nodes and arcs correspond to the set of computations and their data dependencies. By assigning execution times to the nodes and branching probabilities to the exit paths, a program graph is formed. The characteristic equations generated from the program graph are used to predict the intrinsic parallelism in high level programs. This paper presents a method of forming program graphs from the compiler generated tables produced during the translation of the high level program to the data flow language program.

This paper is organized into four sections. The first section presents an overview of the high level language used and the compiler recognized syntactic units of the high level language constructs. The second section presents the estimation of execution times for each of the high level constructs and the corresponding program graphs. The third section presents the procedure to form the program graph for the entire program. The fourth section presents an example program, compiler generated tables and the timed program graph along with the estimated characteristic timing equations.

OVERVIEW OF THE HIGH LEVEL LANGUAGE

The high level language chosen has expressive power similar to Algol 60 (8). Initially, however, the number of data types has been limited to real, integer, boolean, and array. A limited input/output formatting capability is provided along with implied iterative input/output. The language is procedure oriented and eliminates side effects in procedures and functions by allowing changes through the parameter list only, thereby eliminating all global identifiers. The data flow analysis is simplified to some extent by providing the directionality attribute for the argument list and the formal parameters. All the control constructs are forced to have a single entry and exit. The high level constructs consist of syntactic units which are recognized by the compiler. The procedure declaration consists of the syntactic units procedure and end. Similarly the function declaration is made up of function and end syntactic units. The end is the logical end of the procedure and function and it carries pointer information which is useful in the data flow analysis and program graph formation. Input, output, assignment, and procedure call are some of the other syntactic units recognized by the compiler. The syntactic units for the if - then - else construct are:

```
if
 condition
then
close (for then)
else
close (for else)
close (for if)
```

The syntactic units for while-do and repeat-until constructs are:

```
while
  condition
close
repeat
  condition
 close
```

The close syntactic unit serves the purpose of a right paranthesis and contains information to locate the head of the high level construct.

COMPILER GENERATED TABLES, NODE EXECUTION TIME, AND PROGRAM GRAPHS

Compiler Generated Tables

The three stages in the data flow language compilation process are parsing, flow analysis, and code generation. The IFT is formed during the parsing and flow analysis stages. Figure 1a. shows the template for the compiler constructed tables. The statement type field is used in the formaiton of the nodes in the data flow graph and each entry corresponds to one of the compiler recognized syntactic units. The expression tree field (possibly null) is a pointer to the parse tree. The inset and outset fields contain pointers to the inset list and outset list, respectively. The inset list and the outset list indicate the right and left context identifiers. The inset and outset for IFT entries If, Then, Else, Repeat, and While are the union of the insets and outsets of the individual statements in the body of the constructs. The forward link field is the sequential ordering of the entries in the IFT consistent with the ordering of the program text. The backward link field, with the exception of the close entry in the conditional and iteration constructs, points to the previous entry in the table. For the close entry, the backward link is to the start of the corresponding conditional or iteration construct. Figure 1b. shows the linking of <u>uses</u> and <u>definition</u> of an identifier. This linking shows the location where a value for an identifier is produced and where it is used.

Node Execution Times and Program Graphs

Figure 2 shows the program graph for the high level constructs. Each one of the nodes represents a syntactic unit recognized by the compiler. T_s and T_p stand for the sequential and parallel node execution times calculated based on the base machine operation times, in this case the data flow architecture.

In the data flow language (3) a procedure call results in forwarding the procedure name and argument structure as operands. Within the procedure, before the body is executed, the arguments are removed from the argument structure using select operations. Also, before returning to the calling procedure a result structure is formed using an append operation. This is analogous to the prologue and epilogue. The number of select and append operations depends upon the number of identifiers in the parameter list. Therefore, the procedure syntactic unit which is the head of a procedure forms a node in the graph and the node execution time is derived as shown in Figure 2a., where m and n are the number of identifiers in the inset and outset of the procedure. The t(select) and t(append) are the base machine operation time for the operators select and append. In calculating T_p, all the m select operations can be done in parallel whereas the n append operations are done sequentially (8,9). The node body represents the body of the procedure and the end node is the logical end of the procedure. The end node is assigned an execution time of zero because there is no corresponding data flow operator for it. The function definition node is formed in a way similar to the procedure definition node except that the outset has only one identifier.

The input/output operations are assumed to be on a sequential file. The input operation consists of a read machine operation, which supplies a structure consisting of the data value and the file, followed by select operations to select the data value and the file of the structure (8). The output operation performs a write machine operation which returns the file as a result. The assignment statement is represented by a node and T_p and T_s are derived from the parse tree maintained by the compiler by assigning execution times to the operators.

The program graph for the conditional construct is shown in Figure 2f. The nodes if, then, else and close do not have an equivalent data flow operator, therefore, they are assigned zero node execution time. The node execution time for the cond node is evaluated from the conditional expression. The nodes 1 to K and 1 to J are the body of the then and else part of the conditional construct respectively. The iterative constructs are shown in Figure 2g and 2h and the only difference between them is that the conditional node forms part of the body in repeat...until construct as compared to the while construct.

The transitive representation of the program graph is used for the iterative constructs and this is obtained by removing the feedback arc and adjusting the node execution time by multiplying each of the nodes in the loop body by the number of loop iterations. Figure 2i is the program graph for the nonrecursive procedure call. The procedure call consists of an apply machine operation. The apply operation sends the argument structure to the called procedure and receives a result structure from the called procedure. The argument structure is formed by appending the inset values using the append machine operation. Similarly the individual values from the result structure is selected to be sent to the successor operators. Therefore the node execution time is made up of the time to execute the append, apply, select machine operations, and the time taken to execute the procedure body.

PROCEDURE TO FORM A PROGRAM GRAPH FROM COMPILER GENERATED TABLES

Figure 3 shows a high level language program and figure 4 is the IFT generated by the compiler for the high level language program. The field, tree contains pointers to the parse trees and the fields inset and outset are pointers to the inset and outset whose format and connection is shown in Figure 1b. The forward link is used to visit each entry in the IFT while forming the program graph for the entire program.

The syntactic units recognized by the compiler can be divided into three groups according to the way in which they are used to form nodes in the program graph. The first group consists of Input, Output, Assignment, and Procedure invocation. These constructs use the use list of the outset to form successor node(s). The use list is a list of successor location(s) in the program where the value produced for an identifier in the outset is used. For example, consider the node 5 in Figure 5. This node corresponds to entry 5 in the IFT and the outset pointer β_3 points to the outset elements xn and inf. The successor node(s) to node 5 are those which use the identifiers xn and inf. This information is contained in use list. The use list of interest can be located by following the pointer γ in the outset list (see Figure 1b). The identifiers xn and inf will appear in the inset list of the successor node(s). To identify these node(s), the pointer mark α in the use list is used (Fig. 1b).

The second group consists of Procedure definition, Function definition, Repeat, While, and If - Then - Else. These constructs use the use list of the inset to form successor nodes for statements inside the body of the construct. In a feedback data flow processor the body of the conditional and while constructs are executed after evaluation of the predicate. Therefore the successor nodes are formed from the condition node in the program graph using the inset of the if or while entry in the IFT. In the program graph in Figure 5, node 12 stands for the while entry in the IFT. The conditional node is formed by following the forward link. Once the conditional node is formed, the successor nodes are formed using the inset (α_{10}) of the while entry. In the process, any multiple arcs formed are removed. The node execution times T_p and T_s are derived from the tree pointed to by the pointer t_{13}, (Figure 4) assuming unit execution time for all operations.

The third group, consisting of Close and End, is used to form special nodes in the program graph. The Close syntactic unit is used to delimit the while, repeat, and the conditional constructs. The backward link of the close entry points to the corresponding if, while or repeat entry in the IFT. The successor nodes to the above construct are formed using the outset of if, while or repeat entry and are connected to the corresponding close node. In the case of procedure and function definitions the body of the procedure is delimited by the end. The forward pointer of the end entry in the IFT points to the next procedure or function. Therefore, all free nodes in a procedure body are joined to the end node and then a new program graph is started for the next procedure.

Figure 5 shows the program graph constructed from the compiler generated tables. The sequential and parallel characteristic timing equations are constructed as functions of the number of loop iterations and branching probabilities. This construction requires the use of some algorithm to traverse the paths in the program graph (2). The characteristic timing equations for the program graph in Figure 5 are:

$$
\begin{aligned}
T_s &= \text{sum of sequential node execution times} \\
&= 1 + 1 + 1 + 1 + 1 + 4 + 3 + n + 2n + 4n + \alpha_1(3n + n) + \alpha_2(n + 5n) + 11 \\
&\quad + 1 \\
&= 24 + 11n + 4n\alpha_1 + 6n\alpha_2
\end{aligned}
$$

$$
\begin{aligned}
T_p &= \text{parallel execution time} \\
&= \max(1,1,1,1,1,3,3+2) + \max(n+2n,\ n+4n,\ n+4n+\alpha_1(\max(3n,n)) \\
&\quad + \alpha_2\ (\max(3n,n))) + 9 + 1 \\
&= 15 + 5n + 3n\alpha_2 + 3n\alpha_2
\end{aligned}
$$

The ratio of T_s to T_p is defined as the intrinsic parallelism.

To obtain a significant reduction in the number nodes in the program graph, an approximate program graph representation was used in the above method. This approximate program graph representation introduces errors in the characteristic timing equations. A reduction in the parallel execution time can be obtained by producing a balanced parse tree by the compiler. But this error can be eliminated by using tree balancing algorithms given in (5). Another source of error is the inability to display the partial overlap in computations between some data dependent nodes. This error can only be eliminated by using a detailed program graph representation. In the case of conditional and repeat constructs (see Figure 2f and 2i) the data values are collected at the close node before being passed on to the successor node. This representation is not consistent with data flow architecture and thereby introduces errors in the timing equations.

CONCLUSIONS

This paper has presented a method of estimating intrinsic parallelism present in high level language programs at compile time using a program graph model. Techniques to form program graphs and node execution times for well-structured high level constructs are also presented assuming an underlying feedback data flow processor. Also a method to generate the program graph for a given high level program from the compiler generated tables is illustrated. This technique may be used to model systems which satisfy a partial order relationship. This method may also be extended to predict the resources required to execute programs.

REFERENCES

1. Arvind and Gostelow, K. P., "A computer capable of exchanging proces- processor elements for time." Computer Science TR 77, University of California, Irvine, January 1977.
2. Aho, A. V., Hopcroft, J. E. and Ullman, J. D., "The design and analysis of computer algorithms", Addison - Wesley Publishing Company, 1975.
3. Dennis, J. B., "First version of a data flow procedure language." MIT project MAC TM 61, May 1975.
4. Dennis, J. B. and Misunas, D. P. "A preliminary architecture for a basic data flow processor." MIT Project MAC CSG memo 102, August 1974.
5. Kuck, D. J. "A survey of parallel machine organization and programming". ACM Computing Surveys 9", March 1977.
6. Martin, D. E. and Estrin, G., "Models of computational systems - cyclic to acyclic graph transformations", IEEE Transactions on Electronic Computers, Vol. EC-16, No. 1, Feb., 1967.
7. Martin, D. E. and Estrin, G., "Path length computations on graph models of computations." IEEE Transactions on Computers, Vol. C-18, No. 6, June 1969.

8. Oldehoeft, A. E., et.al., "Translation of high level programs to data flow and their simulated execution on a feedback interpreter." Computer Science Technical Report 78-2, Iowa State University, 1978.

9. Oldehoeft, A. E. et.al., "Measurement of parallelism in computer programs through analysis of program graph." Presented in the 1st European Conference on Parallel and Distributed Processing, Feb., 1979.

10. Rumbaugh, J., "A data flow multiprocessor." IEEE Transactions on Computers, 26, Feb. 1977.

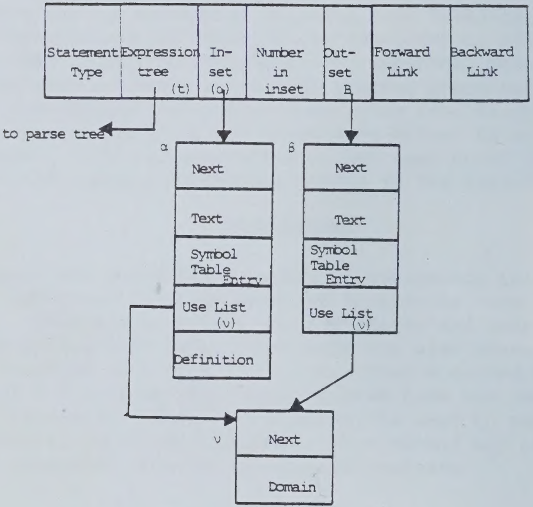
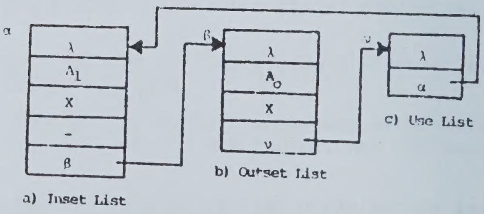

Statement Type
Expression tree (t)
In-set (α)
Number in inset
Out-set β
Forward Link
Backward Link
to parse tree
α
Next
Text
Symbol Table Entry
Use List (ν)
Definition
β
Next
Text
Symbol Table Entry
Use List (ν)
ν
Next
Domain

Figure 1a. Compiler generated tables


α
λ
A_1
X
-
β
β
λ
A_0
X
ν
ν
λ
α

a) Inset List

b) Outset List

c) Use List

Figure 1b Linking uses and definition of an identifier

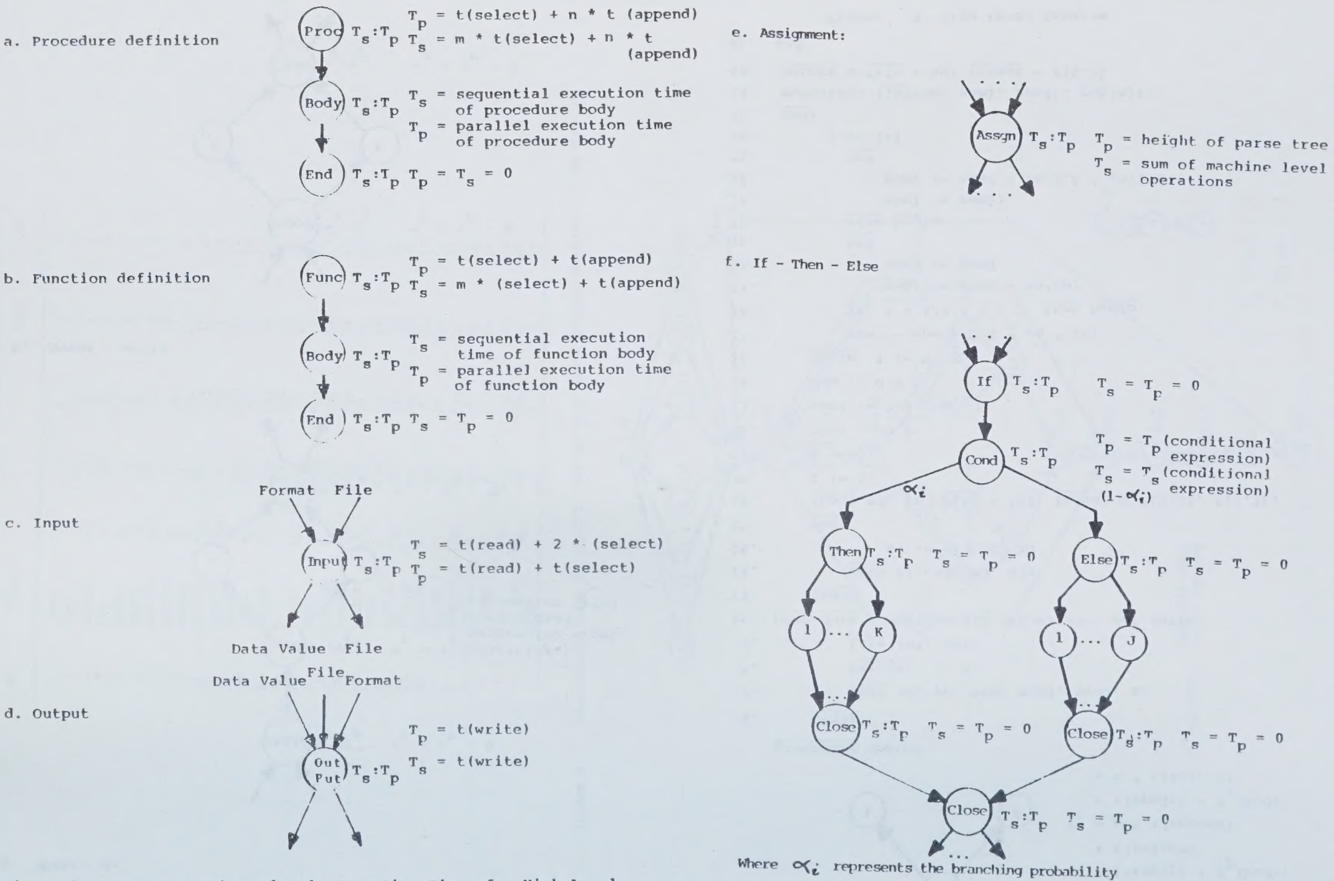
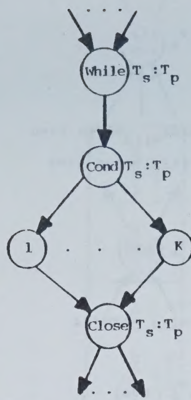

a. Procedure definition
Proc
$T_s : T_p$
T_p = t(select) + n * t (append)
T_s = m * t(select) + n * t (append)
Body
$T_s : T_p$
T_s = sequential execution time of procedure body
T_p = parallel execution time of procedure body
End
$T_s : T_p$
$T_p = T_s = 0$
b. Function definition
Func
$T_s : T_p$
T_p = t(select) + t(append)
T_s = m * (select) + t(append)
Body
$T_s : T_p$
T_s = sequential execution time of function body
T_p = parallel execution time of function body
End
$T_s : T_p$
$T_s = T_p = 0$
c. Input
Format
File
Input
$T_s : T_p$
T_s = t(read) + 2 * (select)
T_p = t(read) + t(select)
Data Value
File
d. Output
Data Value
File
Format
Out Put
$T_s : T_p$
T_p = t(write)
T_s = t(write)
e. Assignment:
Assgn
$T_s : T_p$
T_p = height of parse tree
T_s = sum of machine level operations
f. If - Then - Else
If
$T_s : T_p$
$T_s = T_p = 0$
Cond
$T_s : T_p$
$T_p = T_p$(conditional expression)
$T_s = T_s$(conditional expression)
α_i
$(1-\alpha_i)$
Then
$T_s : T_p$
$T_s = T_p = 0$
Else
$T_s : T_p$
$T_s = T_p = 0$
1
K
1
J
Close
$T_s : T_p$
$T_s = T_p = 0$
Close
$T_s : T_p$
$T_s = T_p = 0$
Close
$T_s : T_p$
$T_s = T_p = 0$
Where α_i represents the branching probability

Figure 2. Program Graphs and Node Execution Times for High Level Constructs.

g. While - do



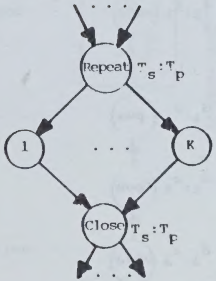
While: $T_s = T_p = 0$

Cond: $T_p = T_p$(conditional expression tree)

$T_s = T_s$(conditional expression tree)

Close: $T_s = T_p = 0$

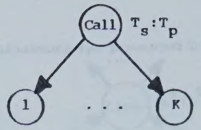
h. Repeat - until:



Repeat: $T_s = T_p = 0$

Close: $T_s = T_p = 0$

i. Procedure call:



$$T_p = m * t(\text{append}) + t(\text{apply}) + T_p(\text{body}) + t(\text{select})$$

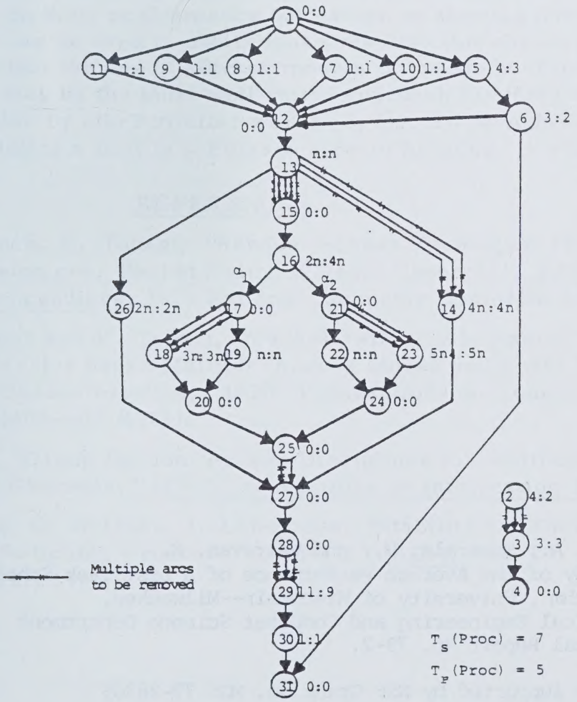
$$T_s = m * t(\text{append}) + t(\text{apply}) + T_s(\text{body}) + n * t(\text{select})$$

```
    procedure mapout
6.      begin
7.          real xn, yn, sum, suml, sum2, s:
8.          integer  I, N;
9.          file inf, opf;
10. procedure summation (in (s1,s2,s3), out (s4))
12.     begin
14.         real s1, s2, s3, s4;
16.             s4 := s1 + s2 + s3
17.     end;
19.     input xn, yn  file = inf  format = F(5,3), F(5,3);
20.     I := 1;
21.     N := 5;
22.     suml := 0.0
23.     sum2 := 0.0
24.     sum := 0.0
26.     while  I <= N  do
27.         sum := sum + 2.0 * xn * yn;
28.         if  I - I/2 * 2 = )  then begin
29.             suml := suml + xn*yn;
30.             sum2 := sum2
31.         end
32.         else begin
33.             suml := suml;
34.             sum2 := sum2 + xn/2.0 * yn/2.0
35.         end
36.       I := I+1
37. end;
39. summation (in(sum, suml, sum2), out(s));
40. output s file = opf format = f(5.3)
41. end
```

Figure 3 High Level Program

Serial No.	Statement Type	Tree	Inset	Outset	Forward Link	Backward Link
1	Proc	t_1	$\wedge$	$\wedge$	5	$\wedge$
2	Proc	t_2	α_1	β_1	3	$\wedge$
3	Assign	t_3	α_4	β_2	4	2
4	END	$\wedge$	α_7	$\wedge$	$\wedge$	3
5	Input	t_5	α_8	β_3	6	1
6	Input	t_6	α_9	β_5	7	5
7	Assign	t_7	$\wedge$	β_7	8	6
8	Assign	t_8	$\wedge$	β_8	9	7
9	Assign	t_9	$\wedge$	β_9	10	8
10	Assign	t_{10}	$\wedge$	β_{10}	11	9
11	Assign	t_{11}	$\wedge$	β_{11}	12	10
12	While	$\wedge$	α_{10}	β_{12}	13	11
13	Cond	t_{13}	α_{17}	$\wedge$	14	12
14	Assign	t_{14}	α_{19}	β_{16}	15	13
15	If	$\wedge$	α_{22}	β_{17}	16	14
16	Cond	t_{16}	α_{27}	$\wedge$	17	15
17	Then	$\wedge$	α_{28}	β_{19}	18	16
18	Assign	t_{18}	α_{32}	β_{21}	19	17
19	Assign	t_{19}	α_{35}	β_{22}	20	18
20	Close	$\wedge$	α_{36}	$\wedge$	21	19
21	Else	$\wedge$	α_{38}	β_{23}	22	17
22	Assign	t_{22}	α_{42}	β_{25}	23	21
23	Assign	t_{23}	α_{43}	β_{26}	24	22
24	Close	$\wedge$	α_{46}	$\wedge$	25	23
25	Close	$\wedge$	$\wedge$	$\wedge$	26	15
26	Assign	t_{26}	α_{48}	β_{27}	27	25
27	Close	$\wedge$	α_{49}	$\wedge$	28	26
28	Close	$\wedge$	$\wedge$	$\wedge$	29	12
29	Call	t_{29}	α_{53}	β_{28}	30	28
30	Output	t_{30}	α_{56}	β_{29}	31	29
31	End	$\wedge$	$\wedge$	$\wedge$	2	30

Figure 4. Intermediate form table (IFT) for the high level program produced by the compiler


1 0:0
11 1:1
9 1:1
8 1:1
7 1:1
10 1:1
5 4:3
6 3:2
12 0:0
13 n:n
15 0:0
16 2n:4n
α_1
α_2
26 2n:2n
17 0:0
21 0:0
14 4n:4n
18 3n:3n
19 n:n
22 n:n
23 5n:5n
20 0:0
24 0:0
25 0:0
27 0:0
28 0:0
29 11:9
30 1:1
31 0:0
2 4:2
3 3:3
4 0:0
Multiple arcs to be removed

$T_S(\text{Proc}) = 7$

$T_P(\text{Proc}) = 5$

Figure 5. Timed Program Graph Generated from Compiler Produced Tables

A STUDY OF THE AVERAGE PERFORMANCE OF A LIST TASK SCHEDULING ALGORITHM+

A. BASHIR, V. SUSARALA, and K. VAIRAVAN
University of Wisconsin--Milwaukee
Milwaukee, Wisconsin

Many algorithms have been designed for parallel scheduling of task systems for multiprocessor applications. One such algorithm is the level (Hu) algorithm. The level algorithm although very simple, is not guaranteed to generate optimal schedules except for the special class of task systems that can be represented by a tree or a forest. In this paper we study the average or expected performance of this algorithm using statistical tools on randomly generated task graphs and an arbitrary number of processors. The results obtained indicate that this algorithm has a very good performance. Indeed it nearly always has given guaranteed optimal schedules for the graphs tested. (1)

---

(1) Bashir, A., Susarala, V., and Vairavan, K.
"A Study of the Average Performance of a List Task Scheduling Algorithm", University of Wisconsin--Milwaukee, Electrical Engineering and Computer Science Department Technical Report No. 79-2.

+ Partially supported by NSF Grant No. MCS 77-28305

A SIMPLE ANALYSIS OF THE CSMA PROTOCOL

ANTHONY EPHREMIDES AND EROL GELENBE
University of Maryland Universite' de Paris-Sud

ABSTRACT

The carrier-sense-multiple access protocol (CSMA) is well suited for use in packet radio channels. In this paper a simple, yet rigorous and correct, analysis is presented that in addition to predicting the performance of the protocol enhances its understanding.

Introduction

The CSMA protocol has been used in several packet-radio channel environments and received in [1, 2] a rather involved analytical treatment. In this paper we employ a different approach to the analysis of the performance of this protocol by utilizing a simple Markov chain model that allows us to derive the throughput results and at the same time to display clearly the mechanism of the markings of this scheme. Our assumptions are similar to the ones made usually in the analysis of multiple access protocols (see for example [1, 2]). We treat the 1-persistent, p-persistent, and non-persistent cases. Our emphasis is not so much in comparing these methods or displaying the already known effects of propagation delay on their performance but rather on showing how a simple analytical tool can be used to obtain these results, thereby strengthening the suggestion that the use of Markov models in the study of multiple access schemes may be the most well suited approach for their analysis, as evidenced also by other similar works [3, 4]. To paraphrase T. Berger, one should stabilize a boat in a Poisson stream by tying it with a Markov chain!

REFERENCES

[1] L. Kleinrock, F. Tobagi, "Random Access Techniques for Data Transmission over Packet Switched Radio Channels", AFIPS Conference Proceedings, 1975 National Computer Conference, 44, 197-201.

[2] L. Kleinrock and F. Tobagi, "Packet Switching in Radio Channels: Part I - Carrier Sense Multiple-Access Modes and Their Throughput Delay Characteristics," IEEE Transactions on Communications, COM-23, 1400-1416 (1975).

[3] T. Rubin, "Group Random Access Disciplines for Multi-Access Broadcast Channels," IEEE Transactions on Information Theory, 1978.

[4] C. Fayolle, G. Gelenbe, I. Labetoulle, "Stability and Optimal Control of Packet Switching Broadcast Channel," JACM, vol. 24, no. 3, pp. 375-386, July 1977.

ON THE NP-COMPLETENESS OF A REAL-TIME SCHEDULING PROBLEM WITH TWO TYPES OF MACHINES†

KAZUO NAKAJIMA AND S. LOUIS HAKIMI
Department of Electrical Engineering and Computer Science
Northwestern University
Evanston, Illinois 60201

ABSTRACT

Consider a set $\mathcal{T}$ of tasks and an unlimited number of machines of two types -- some fast and expensive and the others slow and inexpensive -- which are capable of executing these tasks. Each task $T_i \in \mathcal{T}$ has an integer release time $r_i \geq 0$, an integer deadline $d_i > 0$ and integer processing times p_i and $p_i' > 0$ by a fast and a slow machines, respectively. Assume $p_i < p_i' = d_i - r_i$ for all $T_i \in \mathcal{T}$. We show that the problem of deciding whether there is a machine scheduling that executes all tasks in $\mathcal{T}$ within a designated cost is NP-complete in the strong sense even when $p_i'/p_i = \alpha$ (with α being a fixed integer) for all $T_i \in \mathcal{T}$. Some variations of the problem are shown to be NP-complete in the strong sense. A polynomially solvable case of the problem is also presented.

1. INTRODUCTION

Let $\mathcal{T} = \{T_1, T_2, \ldots, T_n\}$ be a set of n independent tasks. Momentarily, let us assume that we have an unlimited number of identical machines. Each task $T_i \in \mathcal{T}$ has a specified starting time $s_i' \geq 0$ and a processing (execution) time $p_i' > 0$, and must be executed without interruption. A machine can process any task, but no more than one task at a time. Once a machine has completed the processing of a task, it can immediately begin processing another task. The problem is to find the least number of machines and their schedules to process all tasks in $\mathcal{T}$. Several $O(n \log n)$ algorithms for this problem have been reported [4-7,9] and it is also known [4] that $O(n \log n)$ is the lowest possible order of complexity for finding a solution for this problem.

In this paper, we will consider a natural generalization of this problem by allowing two types of machines -- fast machines and slow machines -- which are capable of executing any task in $\mathcal{T}$. Let $r_i \geq 0$ and $d_i > 0$ be the integer release time and integer deadline for task $T_i \in \mathcal{T}$. Let $p_i > 0$ and $p_i' > 0$ be the integer processing times of task $T_i \in \mathcal{T}$ by a fast and a slow machines, respectively. We assume $p_i < p_i' = d_i - r_i$ for all $T_i \in \mathcal{T}$; that is, a slow machine must start executing task $T_i \in \mathcal{T}$ at time r_i to meet its deadline. However, a fast machine may start executing task $T_i \in \mathcal{T}$ at any integer time s_i in the interval $[r_i, d_i - p_i]$ to meet its deadline. Once a machine of either type has completed the processing of a task, it can immediately start processing another task; that is, the processing interval of task $T_i \in \mathcal{T}$ is a semi-open interval $[s_i, s_i + p_i)$ by a fast machine, or $[r_i, r_i + p_i')$ by a slow machine.

Let c and c' be the integer costs of a fast and a slow machines, re-

---

†This work was supported by the National Science Foundation Grant ENG79-09724.

spectively. We assume $c > c' > 0$. Let m and m' be the numbers of fast and slow machines, respectively, used to process all tasks in $\mathcal{T}$. Then we will say that there is an (m, m')-machine assignment for $\mathcal{T}$. The cost of this assignment is $mc + m'c'$. Let $c_o > 0$ be a designated integer cost. Then the "controlled cost real-time scheduling" problem is stated as follows:

CCRS: Is there an (m, m')-machine assignment for $\mathcal{T}$ with $mc + m'c' \leq c_o$?

In the next section we will prove that the CCRS problem is NP-complete in the strong sense [2,3] even when $p_i'/p_i = \alpha$ (with α being a fixed integer) for all $T_i \in \mathcal{T}$. We will also consider a special case of the CCRS problem where both numbers of fast and slow machines are limited (bounded by constants). It will be shown that even in this case the CCRS problem is NP-complete in the strong sense when $p_i'/p_i = \beta$ (with β being an arbitrary integer) for all $T_i \in \mathcal{T}$. We will then briefly examine a polynomially solvable case of the CCRS problem. In the last section we will review the strong NP-completeness of two more restricted cases of the CCRS problem and a polynomially solvable case [9,10].

2. NP-COMPLETENESS RESULTS

We will first show that the CCRS problem is NP-complete in the strong sense when $p_i'/p_i = \alpha$ (with α being a fixed integer) for all $T_i \in \mathcal{T}$. Our proof will be based on a transformation of the 3-PARTITION problem to the CCRS problem. The 3-PARTITION problem is known to be NP-complete in the strong sense [1] and may be stated as follows:

3-PARTITION: Given positive integers q and b, and a set* of integers $A = \{a_1, a_2, \ldots, a_{3q}\}$ with $\sum_{i=1}^{3q} a_i = qb$ and $b/4 < a_i < b/2$ for $1 \leq i \leq 3q$; can A be partitioned into q disjoint 3-element sets $A_1, A_2, \ldots, A_q$ such that $\sum_{a \in A_j} a = b$ for $1 \leq j \leq q$.

Theorem 1. The CCRS problem is NP-complete in the strong sense when $p_i'/p_i = \alpha$ (with α being a fixed integer) for all $T_i \in \mathcal{T}$.

Proof. The problem clearly belongs to NP. Let $A = \{a_1, a_2, \ldots, a_{3q}\}$ with integers q and b be an instance of the 3-PARTITION problem. We construct a real-time task system $\mathcal{T}(A)$ to correspond to the set A as follows:

$\mathcal{T}(A) = X \cup Y$,

where $X = \{T_i \mid 1 \leq i \leq 3q,\ r_i = 0,\ p_i = 4a_i,\ p_i' = 16a_i,\ d_i = r_i + p_i' = 16a_i\}$

and $Y = \{T_i \mid 3q+1 \leq i \leq 7q,\ r_i = 4b,\ p_i = b,\ p_i' = 4b,\ d_i = r_i + p_i' = 8b\}$.

We note that $p_i'/p_i = 4$ for all $T_i \in \mathcal{T}(A)$. Let c and c' be the costs of a fast and a slow machines, respectively, such that $c' < c < 7c'$. Let $c_o = qc$. We note that the above transformation is clearly a pseudo-polynomial transformation [2,3]. Thus we need only show that there is an (m, m')-machine assignment for $\mathcal{T}(A)$ with $mc + m'c' \leq c_o$ if and only if A can be partitioned into q disjoint 3-element sets $A_1, A_2, \ldots, A_q$ such that $\sum_{a \in A_j} a = b$ for $1 \leq j \leq q$.

---

* We allow a set to have duplicate elements.

Suppose that there is a partition $< A_1, A_2, \ldots, A_q >$ of A into 3-element sets such that $\sum_{a \in A_j} a=b$ for $1 \le j \le q$. We will show that all tasks in $\mathcal{T}(A)$ can be processed by q fast machines. We note that we can assign all tasks in Y to q fast machines in such a way that each fast machine executes exactly four tasks during the interval [4b, 8b). This assignment leaves each fast machine to have an idle interval [0, 4b). Then we assign the three tasks in the set $X_j = \{T_i | p_i = 4a_i, a_i \in A_j\}$ to the j-th fast machine during this idle interval, for $1 \le j \le q$. This can be done, because $\sum_{T_i \in X_j} p_i = \sum_{a_i \in A_j} 4a_i = 4b$ and $r_i = 0$ and $d_i > 4b$ for all $T_i \in X_j$, $1 \le j \le q$. Therefore, the cost of this assignment is qc and it is within the cost c_o.

Suppose, for the other direction, that there is an (m, m')-machine assignment for $\mathcal{T}(A)$ with $mc + m'c' \le c_o$. Since $d_i = 16a_i > 4b$ for all $T_i \in X$ and $r_j = 4b$ for all $T_j \in Y$, the slow machine processing intervals of all 7q tasks in $\mathcal{T}(A)$ overlap (intersect). Suppose that k ($< q$) fast machines are used. Since $b/4 < a_i < b/2$ for $1 \le i \le 3q$, we have $b < p_i = 4a_i < 2b$ and $4b < d_i = 16a_i < 8b$ for all $T_i \in X$, and $p_j = b$ and $d_j = 8b$ for all $T_j \in Y$. This implies that the total number of tasks assigned to a fast machine is no greater than 8. However, as $r_j = 4b$ for all $T_j \in Y$, in fact the total number of tasks assigned to a fast machine is at most 7. Thus k fast machines can process at most 7k tasks and hence at least 7(q - k) tasks must be processed by slow machines. However, as the slow machine processing intervals of all these tasks overlap, we need at least 7(q - k) slow machines to process them. Therefore, the cost of this assignment $C \ge kc + 7(q - k)c'$. But as $c < 7c'$, we have $C > kc + (q - k)c = qc = c_o$. Thus, if there is an (m, m')-machine assignment with $mc + m'c' \le c_o$, precisely q fast machines must be used. Furthermore, as $c_o = qc$, no slow machine can be used.

Since $|Y| = 4q$ and $p_i = b$, $r_i = 4b$ and $d_i = 8b$ for all $T_i \in Y$, each fast machine must be entirely busy executing exactly four tasks in Y during the interval [4b, 8b). Therefore, all the remaining 3q tasks in X must be assigned to fast machines during the interval [0, 4b). Since $b < p_i = 4a_i < 2b$ for all $T_i \in X$ and $|X| = 3q$, exactly three tasks in X must be assigned to each fast machine. However, we have $\sum_{T_i \in X} p_i = \sum_{a_i \in A} 4a_i = 4qb$ and hence each fast machine must be entirely busy executing three tasks in X during the interval [0, 4b). This implies that the total processing time of the three tasks in X which are executed by each fast machine must be equal to 4b. Let T_{j1}, T_{j2} and T_{j3} be the three tasks in X which are executed by the j-th fast machine. Then $\sum_{i=1}^{3} p_{ji} = 4b$ and hence $\sum_{i=1}^{3} a_{ji} = b$. Let $A_j = \{a_{j1}, a_{j2}, a_{j3}\}$. Then, we have a desired partition of A into 3-element sets $A_1, A_2, \ldots, A_q$ such that $\sum_{i=1}^{3} a_{ji} = b$ for $1 \le j \le q$. This completes the proof. □

We note that the above proof in fact shows that the CCRS problem in the case when $p_i'/p_i = \alpha$ (with α being a fixed integer) for all $T_i \in \mathcal{T}$ is still NP-complete in the strong sense even in the situation where the number of available slow machines is limited (bounded by a constant). We also note that in the above proof $1 < c/c' < 7$ and $p_i'/p_i = 4$ for all $T_i \in \mathcal{T}(A)$. Thus, even if $c/c' = p_i'/p_i$, the problem is still NP-complete in the strong sense.

In the following we will show that the CCRS problem is NP-complete in the strong sense even in the situation where both the numbers of fast and slow machines are limited when $p_i'/p_i = \beta$ (with β being an arbitrary integer) for all $T_i \in \mathcal{T}$.

Theorem 2. The CCRS problem is NP-complete in the strong sense even when limited numbers of fast and slow machines are available and $p_i'/p_i = \beta$ (with β being an arbitrary integer) for all $T_i \in \mathcal{T}$.

Proof. This proof is a modification of the one given in [3,pp.102-103]. Let $A = \{a_1, a_2, \ldots, a_{3q}\}$ with integers q and b be an instance of the 3-PARTITION problem such that $\sum_{i=1}^{3q} a_i = qb$ and $(b + 1)/4 < a_i < b/2$** for $1 \leq i \leq 3q$. We construct a task system $\mathcal{T}(A)$ as follows:

$$\mathcal{T}(A) = X \cup Y_1 \cup Y_2 \cup \ldots \cup Y_{q-1} \cup Z,$$

where $X = \{T_i \mid 1 \leq i \leq 3q,\ r_i = 0,\ p_i = 4qa_i,\ p_i' = 16q^2a_i,\ d_i = r_i + p_i' = 16q^2a_i\}$,

$$Y_j = \{T_i \mid 4qj - q + 1 \leq i \leq 4qj + 3q,\ r_i = 4q(bj + j - 1),\ p_i = 1,\ p_i' = 4q,\ d_i = r_i + p_i' = 4q(b + 1)j\},\ \text{for } 1 \leq j \leq q-1,$$

and $$Z = \{T_i \mid 4q^2 - q + 1 \leq i \leq 4q^2 + 3q,\ r_i = 4q(bq + q - 1),\ p_i = bq - q + 1,\ p_i' = 4q(bq - q + 1),\ d_i = r_i + p_i' = 8q^2b\}.$$

We note that $p_i'/p_i = 4q$ for all $T_i \in \mathcal{T}(A)$. Let c and c' be the costs of a fast and a slow machines, respectively, such that $c' < c < 7qc'$. Let $c_o = c$. We note that the above transformation is clearly a pseudo-polynomial transformation. Thus, we need only show that there is an (m, m')-machine assignment for $\mathcal{T}(A)$ with $mc + m'c' \leq c_o$ if and only if A can be partitioned into q disjoint 3-element sets $A_1, A_2, \ldots, A_q$ such that $\sum_{a \in A_j} a = b$ for $1 \leq j \leq q$.

Suppose that there is a partition $< A_1, A_2, \ldots, A_q >$ of A into 3-element sets such that $\sum_{a \in A_j} a = b$ for $1 \leq j \leq q$. We will show that all tasks in $\mathcal{T}(A)$ can be processed by a single fast machine. We note that we can assign all tasks in $Y_1, Y_2, \ldots, Y_{q-1}$ and Z to the fast machine. This assignment leaves the fast machine to have q idle intervals $[4q(b+1)(j-1), 4q(bj+j-1))$ for $1 \leq j \leq q$, which are all of length 4qb. Then we assign the three tasks in the set $X_j = \{T_i \mid p_i = 4qa_i, a_i \in A_j\}$ to the j-th idle interval on the fast machine for $1 \leq j \leq q$. This can be done, because $\sum_{T_i \in X_j} p_i = \sum_{a_i \in A_j} 4qa_i = 4qb$ and $r_i = 0$ and $d_i = 16q^2a_i \geq 4q^2b + 4q^2 > r_k$ for all $T_i \in X$ and $T_k \in Z$. Therefore, the cost of this assignment is c, which is equal to c_o.

Suppose, for the other direction, that there is an (m, m')-machine assignment for $\mathcal{T}(A)$ with $mc + m'c' \leq c_o$. Suppose that all tasks in $\mathcal{T}(A)$ are assigned to slow machines. Since all 3q tasks in X and all 4q tasks in each Y_j, $1 \leq j \leq q-1$, overlap, we need at least 7q slow machines to process all tasks in X and Y_j, and the cost of this assignment already exceeds c_o

---

** We have made a slight modification of the range of a_i, $1 \leq i \leq 3q$. However, this does not affect the strong NP-completeness of the 3-PARTITION problem (see, [3, p. 99]).

which is equal to $c < 7qc'$. Therefore, a fast machine must be used to process all tasks in $\mathcal{T}(A)$ within the cost $c_o = c$ and in fact no slow machine can be used.

Since the total processing time of all 4q tasks in each Y_j, $1 \le j \le q-1$, by a fast machine is 4q, and all these tasks in each Y_j must be processed during the interval $[4q(bj+j-1), 4q(b+1)j)$, the fast machine must be entirely busy during each such interval. Similarly we can see that this machine must be entirely busy processing all tasks in Z during the interval $[4q(bq+q-1), 8q^2b)$.

The above task assignments leave q idle intervals $[4q(b+1)(j-1), 4q(bj+j-1))$ for $1 \le j \le q$, which are all of length 4qb. Since $r_i = 0$ and $4q(bj+j-1) < d_i = 16q^2a_i < 8q^2b$ for all $T_i \in X$ and j, $1 \le j \le q$, all tasks in X must be processed during these idle intervals. Furthermore, the total processing time of these tasks by a fast machine is $4q^2b$ and the total idle time during these q intervals is also $4q^2b$. Therefore, the tasks in X must be partitioned into q disjoint subsets $X_1, X_2, \ldots, X_q$ of tasks such that $\sum_{T_i \in X_j} p_i = 4qb$ for $1 \le j \le q$. Furthermore, since $q(b+1) \le p_i = 4qa_i < 2qb$ for all $T_i \in X$, each subsets X_j must contain exactly three tasks, say T_{j1}, T_{j2} and T_{j3}, for $1 \le j \le q$. Let $A_j = \{a_{ji} \mid p_{ji} = 4qa_{ji}, T_{ji} \in X_j, 1 \le i \le 3\}$. Then $\sum_{i=1}^{3} a_{ji} = \sum_{i=1}^{3} p_{ji}/4q = b$. Hence $< A_1, A_2, \ldots, A_q >$ is a desired partition of A. □

The reader should note here that the above two proofs in fact show the NP-completeness of the following scheduling problems.

(1) Given a set $\mathcal{T} = \{T_1, T_2, \ldots, T_n\}$ of tasks with each T_i having an integer processing time $p_i > 0$, an integer release time $r_i \ge 0$, and an integer deadline $d_i > 0$, and m identical processors, is there a schedule on m processors that execute every task $T_i \in \mathcal{T}$ entirely within the interval $[r_i, d_i)$ when $d_i - r_i = \alpha p_i$ and $r_i \in \{0, \gamma\}$ with α and γ being a fixed and an arbitrary integers, respectively.

(2) Given a set $\mathcal{T}$ of tasks defined above and a single machine, is there a schedule on a single machine that executes every task $T_i \in \mathcal{T}$ entirely within the interval $[r_i, d_i)$ when $d_i - r_i = \beta p_i$ with β being an arbitrary integer.

We now wish to briefly examine a polynomially solvable case of the CCRS problem. Suppose we have only one fast machine and an unlimited number of slow machines. We assume $r_i = 0$ for all $T_i \in \mathcal{T}$. Using Moore's algorithm [8], we can find in $O(n \log n)$ time a sequence of tasks which minimizes the number of "late" tasks -- the ones which cannot be completed before their deadlines -- when all tasks in $\mathcal{T}$ are assigned to a fast machine. These late tasks must be assigned to slow machines. Since the slow machine processing intervals of all tasks in $\mathcal{T}$ overlap, these late tasks must be assigned to different slow machines. The above assignment gives a least cost task assignment if exactly one fast machine is used. We note that all tasks in $\mathcal{T}$ may also be processed by n slow machines. Whichever of these two assignments which is of lesser cost is the optimum.

3. A REVIEW OF SOME RESULTS ON THE CCRS PROBLEM

In Section 2, we considered the CCRS problem in the case when each task $T_i \in \mathcal{T}$ must be started for execution by a slow machine at time r_i, but it may be started for execution by a fast machine at any integer time s_i in the "starting time interval" $[r_i, d_i-p_i]$. In this section, we will consider the same CCRS problem in a more restricted case where the starting time interval for a fast machine is also a fixed point for each task $T_i \in \mathcal{T}$. More specifically, we consider the case where any task $T_i \in \mathcal{T}$ must be immediately started for execution by a machine of either type as soon as it is released. Even in this case it can be shown [9,10] that the CCRS problem is NP-complete in the strong sense for the following cases:

(1) $p'_i/p_i = \alpha$ (with α being a fixed integer) for all $T_i \in \mathcal{T}$.

(2) $p_i = \pi$ (with π being an arbitrary integer) for all $T_i \in \mathcal{T}$.

We note, however, that if $p_i=1$ for all $T_i \in \mathcal{T}$, a minimum cost (m,m')-machine assignment can be found in $O(n \log n)$ time [9]. In Table 1, we summarize the complexity results on as well as our conjectures about the CCRS problem.

Table 1. The complexity results on the CCRS problem.

s_i	$s_i \in [r_i, d_i-p_i]$		$s_i = r_i$	
p_i \ p'_i/p_i	no restriction	α	no restriction	α
no restriction	NPC	NPC	NPC	NPC [9]
π	NPC?	?	NPC [10]	?
1	?	P?	O(n log n) [9]	

Notes: s_i: The integer starting time of task $T_i \in \mathcal{T}$ by a fast machine.
NPC: NP-complete in the strong sense.
P? (NPC?): Conjectured to be polynomially solvable (NP-complete in the strong sense).
α (π): A fixed (an arbitrary) integer.

REFERENCES

[1] M. R. Garey and D. S. Johnson, Complexity results for multiprocessor scheduling under resource constraints, SIAM J. Computing 4 (1975), pp. 397-411.

[2] M. R. Garey and D. S. Johnson, "Strong" NP-completeness results: motivation, examples, and implications, J. Assoc. Comput. Mach., 25 (1978), pp. 499-508.

[3] M. R. Garey and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman, San Francisco, CA, 1979.

[4] U. I. Gupta, D. T. Lee and J. Y-T. Leung, An optimal solution for the channel-assignment problems, IEEE Trans. Computers, to appear.

[5] A. Hashimoto and J. E. Stevens, Path cover of acyclic graphs, ILLIAC IV, Document No. 239, Univ. of Illinois, Urbana, IL, 1970.

[6] A. Hashimoto and J. E. Stevens, Wire routing by optimizing channel assignment within large apertures, Proc. 8th Design Automation Workshop, 1971, pp. 155-169.

[7] B. W. Kernighan, D. G. Schweikert and G. Persky, An optimum channel-routing algorithm for polycell layouts of integrated circuits, Proc. 10th Design Automation Workshop, 1973, pp. 50-59.

[8] J. E. Moore, An n job, one machine sequencing algorithm for minimizing the number of late jobs, Management Science, 15 (1968), pp. 102-109.

[9] K. Nakajima and S. L. Hakimi, On certain real-time scheduling problems with two types of machines, submitted for publication, July 1979.

[10] K. Nakajima and S. L. Hakimi, A scheduling problem with two types of machines, in preparation.

PREEMPTIVE SCHEDULING OF PERIODIC, REAL-TIME TASKS†

JOSEPH Y-T. LEUNG and M. L. MERRILL
Department of Electrical Engineering and Computer Science
Northwestern University
Evanston, Illinois 60201

SUMMARY

We consider the problem of preemptively scheduling a set of periodic, real-time tasks on 1 processor. Each task T_i, is characterized by (e_i, d_i, p_i, s_i). In this characterization, T_i is requested initially at time s_i, and thereafter at times $s_i + kp_i$, $k=1,2,\ldots$. The execution time required by T_i is e_i, and a deadline for T_i occurs d_i time units after each request, by which time T_i must have completed its execution. A task system R consists of a collection of n tasks $\{T_i\}_{i=1}^{n}$ and is denoted by $R=(\{T_i\},\{e_i\},\{d_i\},\{p_i\},\{s_i\})$. A schedule for R is said to be feasible if every deadline of each task in R is met. R is said to be schedulable if there exists a feasible schedule for R. A scheduling algorithm is said to be optimal if it produces a feasible schedule for every task system which is schedulable.

In this summary we shall give an optimal scheduling algorithm and an algorithm to decide if a task system is schedulable. Because of space limitation, we shall omit proofs of all theorems as well as many references related to this work. The interested reader is referred to [1] for detailed discussions. The scheduling algorithm to be considered will be called the Slack-Time Algorithm. Before we introduce this algorithm, we need to define the following terminology. Let task T be active †† at time t and let d denote the deadline of the corresponding request of T. If e denotes the remaining processing time needed by T at time t, then the slack-time of T at time t is defined to be d-t-e. The Slack-Time Algorithm schedules, at each instant of time, that active task whose slack-time is the smallest among those unassigned active tasks. If there are k tasks tied for the same slack-time, then they share the processor equally. We have the following result about the Slack-Time Algorithm.

Theorem 1 The Slack-Time Algorithm is optimal.

We now turn to the problem of deciding if a task system is schedulable. The algorithm consists of constructing a schedule using the Slack-Time Algorithm and check to see if all deadlines are met. Before we present this result, we need to introduce the following notation. Let S be a schedule of $R=(\{T_i\},\{e_i\},\{d_i\},\{p_i\},\{s_i\})$. We define the configuration of S for R at time t, denoted by $C_S(R,t)$ to be the n-tuple $(e_{1,t}, e_{2,t}, \ldots, e_{n,t})$, where $e_{i,t}$ is the amount of time for which T_i has executed since its last request up until time t, and $e_{i,t}$ is undefined if $t < s_i$. We let s_R denote $\max\{s_i\}$ and P_R denote the LCM of $\{p_1,\ldots,p_n\}$. We assume that $\min\{s_i\} = 0$. The following result leads to an algorithm for deciding if a task system is schedulable.

Theorem 2 Let S be the schedule of a task system R constructed by the Slack-Time Algorithm. R is schedulable iff (1) all deadlines in the interval $[0,t_2]$ are met in S, where $t_2 = s_R+2P_R$, and (2) $C_S(R,t_1) = C_S(R,t_2)$, where $t_1 = s_R+P_R$.

---

† This research was supported in part by the National Science Foundation under Grant MCS-79-04898.

†† A task is said to be active if it has requested for execution but has not received a processing time equal to its execution time.

References

1 J. Y-T. Leung and M. L. Merrill, "Preemptive Scheduling of Periodic, Real-Time Tasks," T.R. #79-06-FC-02, Dept. of Electrical Engineering and Computer Science, Northwestern University, Evanston, Illinois 60201, July, 1979.

ON THE STRUCTURE OF SETS IN NP AND OTHER COMPLEXITY CLASSES

L.H.LANDWEBER + | R.LIPTON * | E.L.ROBERTSON +
Univ. of Wisconsin | Univ. of Calif. | Indiana Univ.
Madison,WI | Berkeley,CA | Bloomington,IN

ABSTRACT

A simple technique is developed for manipulating the complexity of sets with respect to polynomial time reducibility. One application is the definition of a minimal pair (with respect to polynomial time reducibility) of sets in NP-P. The last section proves that the NP-complete sets are effectively enumerable while NP-P is not.

INTRODUCTION

In this paper, we study the structure of sets in NP. We develop a number of simple tools which facilitate the study of the relative complexity of sets with respect to polynomial time reducibility. The method yields the existence of a minimal pair (with respect to P) of sets $A,B \in NP$ which are not complete and which satisfy, $C \leq_p A$ and $C \leq_p B$ imply $C \in P$. This strengthens earlier results of Ladner[9] (minimal pair- no upper bound) and Machtey[14] (minimal pair- subexponential but not necessarily in NP). In addition, the method can be used to construct arbitrary partial orders of degrees with respect to polynomial time reducibility. Similar results have recently been obtained by Breidbart[15] though his proofs are significantly more complicated because of their dependance on the Recursion Theorem.

The main idea involves an analysis of those input regions on which set membership can be decided in polynomial time. Because of the restrictions of polynomial time reducibility, a bound may then be obtained on the size of such intervals in sets to which a given set can be reduced. This yields the existence of sets in NP which are not complete and which have a variety of other properties.

Recently a number of papers [1,2,5,6,7,11] have dealt with such properties as density and sparseness for sets in NP. A set is p-sparse [2] if there is a polynomial q such that for all n at most q(n) strings of length n are in the set. It is not known whether NP-complete sets can be

---

*Supported in part by NSF Grant MCS79-20409 and the Army Mathematics Research Center.

+Supported in part by NSF Grant MCS76-17323.

This research we facilitated by the use of Theory Net (NSF Grant MCS78-01689).

sparse. The relationship between these results and ours is not clear. By introducing gaps (and thereby easy intervals) into a set a sparser set is obtained (e.g., if one considers distance between set members). Hence our results can be interpreted as imposing limits on how sets can differ with respect to degree of sparseness and still be related by polynomial time reducibility.

Recent results by Meyer and Paterson[6] provide a first step towards comparing the various notions of sparseness and complexity properties. Their paper genralizes and unifies earlier work by P. Berman[2], Fortune[5] and Hartmanis and Berman[7] on sparseness and in addition provides a link to our results. Many open problems remain to be considered. is an area for future research.

We believe that our results and those referenced above are important because they provide intuition with respect to the structural properties of NP-complete sets. They represent an attempt to look more closely at why sets are NP-complete as opposed to which sets are NP-complete. While demonstrating that a large number of problems are NP-complete has served to increase the importance of the P = NP question, it does not appear to have brought us closer to a solution to this problem. Recursion theory is an area thich has benefitted from a similar study involving the structural properties of and the relationship between recursively enumerable sets. This approach has led to the development of powerful tools(e.g., the priority method) and has resulted in a fairly complete understanding of the r.e. sets. Hopefully, further study of the relationship between structure and complexity will lead to a similar state of affairs for NP and other complexity classes.

The remainder of the paper investigates the existence of effective enumerations of devices for recognizing all and only members of certain classes of sets. A class admitting such an enumeration is said to be recursively presentable. See [10] for earlier work in this area for abstract complexity measures. In particular, it is proved that the NP-complete sets are recursively presentable and moreover the devices all operate in nondeterministic polynomial time. Similar results are obtained for other complexity classes (PSPACE, EXP-TIME, etc). On the other hand if $P \neq NP$, the class NP-P is not recursively presentable by any devices which compute characteristic functions. It is open whether NP-P can be presented in terms of devices whose domains are the sets in question. In contrast to NP-complete sets, we show that the complete recursively enumerable sets are not recursively presentable.

2. DEFINITIONS

Notation We use $|x|$ for the length of a string in Σ^*.

We consider two notions of polynomial time reducibility due to Cook [4] and Karp [8], many-one polynomial time reducibility ($\leq_p$) and Turing polynomial time reducibility ($\leq_{pT}$).

Write $A <_p B$, if $A \leq_p B$ but not $B \leq_p A$. Also $A \equiv_p B$ if $A \leq_p B$ and $B \leq_p A$. Similarly define $<_{pT}$ and $\equiv_{pT}$.

A function f is time constructible if there is a Turing machine which on inputs of length n halts in exactly f(n) steps.

A set of functions $\{f_i\}$ is recursively presentable if there is an effective enumeration $M_0, M_1, \ldots$ of Turing machines such that 1) each f_i is computed by some M_j and 2) each M_j computes a member of $\{f_i\}$. Sets are identified with their characteristic functions yielding a definition of recursively presentable for a collection of sets. SAT denotes the satisfiable propositional calculus formulas encoded suitably in an alphabet.

For r increasing, define G[r] by

$$G[r] = \{x \mid r^{\#}(0,n) \leq |x| < r^{\#}(0,n+1) \quad \text{for } n \text{ even}\}.$$

G[r] will be used to introduce r sized gaps into sets.

For $n \in N$, the r-interval at n, I[r,n] is given by

$$I[r,n] = \{y \in \Sigma^* \mid n \leq |y| < r(n)\}$$

A set has r-gaps or gap size r if there are arbitrarily large n such that $y \in I[r,n]$ implies $y \notin B$. The polynomial time algorithm Q is correct for B on arbitrarily large r-intervals if there exist arbitrarily large n such that for all $y \in I[r,n]$, $Q(y) = B(y)$.

A recursive set $B \subset \Sigma^*$ is r-interval easy for some recursive r, if there is a polynomial q and an algorithm having time complexity f which recognizes B such that for infinitely many n, $y \in I[r,n]$ implies $f(|y|) \leq q(|y|)$.

Note that if a set has r-gaps, or is r-interval easy this implies that some polynomial time algorithm is correct on the set for arbitrarily large r-intervals.

3. STRUCTURAL PROPERTIES

Previous studies have revealed a rich degree structure with respect to polynomial time reducibility . In this section we investigate structural properties of sets which result from their position in the reducibility hierarchy. The strict constraints associated with polynomial time reducibility allow us to develop a fairly simple tool which has a number of applications. Analogous questions in classical recursion theory include the relationship between simple and hypersimple sets and various reducibility notions.

The first theorem imposes a maximum easy interval size on sets to which a given set is many-one polynomial time reducible.

Theorem 1. Let A be a recursive set. If $A \notin P$, then there is a recursive r such that for each B satisfying $A \le_p B$, no polynomial time algorithm is correct for B on arbitrarily large r-intervals.

Corollary 2. If $A \notin P$, then there is a maximum gap size for the $\equiv_p$ class of A.

Corollary 3. If $A \notin P$, then there is a maximum gap size for B such that $A \le_p B$.

The fact that a set has r-gaps for large r does not imply that is "easy". By standard complexity theory techniques, recursive sets of arbitrary complexity may be constructed with r-gaps This together with Theorem 1 means that given any set A, there are arbitrarily complex sets with arbitrarily large gaps which are incomparable (with respect to $\le_p$) with A. Conversely, the introduction of large enough (polynomial time recognizable) gaps in a set A guarantees that the resulting set A' satisfies $A' <_p A$. (Theorem 8 of Section 4 provides a more significant application of this technique because the A' above may be in P whereas $A \notin P$.)

In contrast to the above, note that polynomial size gaps can be introduced in SAT with the resulting set $SAT \cap G[p]$ remaining NP-complete.

The following results deal with the existence of gaps for NP-complete sets. Related questions were considered in [1,2,5,7]. Studies of the gap-density properties of NP-complete sets will hopefully increase our intuition with respect to the difficult open problems for this class.

Theorem 4. Assume that a polynomial time algorithm is correct for SAT for arbitrarily large r-intervals and some recursive r. Then there is a polynomial time computable set A such that $SAT \cap A$ is NP-complete but has r-gaps.

Theorem 5. Let B be an NP-complete set, r a recursive function and Q a polynomial time algorithm such that Q is correct for B on arbitrarily large r-intervals. Then there exist NP-complete sets with $f \cdot r \cdot \log$-gaps where f is any function satisfying $f^{-1}(n) > n^k$ a.e. for all k.

The next theorem provides a bound for each $A \notin P$ on the size of intervals for which A can be easy, i.e., on which it can be recognized in polynomial time. Notice that this also yields an upper bound on permissible gaps in non-polynomial sets since recognition is trivial on gaps. This theorem and the one following enable us to strengthen earlier results on the structure of sets in NP and, in addition, provide an elegant and consistent approach for obtaining such results.

Theorem 6. Let $A \notin P$ be recursive. Then there is a recursive r such that no polynomial time algorithm is correct for A on arbitrarily large r-intervals. Hence, no algorithm for A can be r-interval easy (see the remark at the end of Section 2).

Theorem 1 is based on the fact that if B has large easy intervals and $A \leq_p B$, then the transduction from A to B must , on inputs in an easy B-interval, either produce short strings or strings on which a polynomial algorithm is correct for B. A more complicated argument must be used for polynomial time Turing reductions, where more than one question may be asked during a computation and where answers must actually be computed (or somehow simulated) for the computation to proceed. For Turing reducibility, the upper bound on easy interval sizes is not as sharp as for many-one polynomial-time reducibility(by one exponential).

Theorem 7. Let $A \notin P$ be recursive. There is a recursive f such that if $A \leq_{pT} B$ and some polynomial time algorithm is correct for B on arbitrarily large r-intervals, then $r \leq f$ a.e. Hence if such a B is r-interval easy, then $r \leq f$ a.e.

4. APPLICATIONS

In Section 3, we introduced the notion of (easy) gaps and developed a few relationships between easy intervals (gaps) and complexity structures. We now apply this machinery to study the existence of noncomplete sets in NP having certain properties. Unless otherwise specified, we assume throughout that $P \neq NP$.

Theorem 8. [Ladner] Let $B \notin P$ be a recursive set. Then there exists a recursive set A satisfying:

a. $A \notin P$
b. $A \leq_p B$
c. $B \not\leq_{pT} A$

Proof. Let $B \notin P$ be a recursive set. Choose r recursive such that r is time constructible and C is not r-interval easy for any C such that $B \leq_{pT} C$. Let $D = G[r]$. Clearly $D \in P$. Let $A = B \cap D$. Since $B \cap D$ is r-interval easy, $B \not\leq_{pT} B \cap D$ by Theorem 7. Clearly $B \cap D \leq_p B$. Finally, $B \cap D \notin P$ because B is not r-interval easy $(\overline{D} = G[r])$.

Corollary 9. [Ladner] There are sets in NP-P which are not NP-complete.

Observe that by introducing sufficiently large gaps, any NP-complete set can be transformed into a member of NP-P which is not complete. Also, by choice of sufficiently large r, we can diagonalize out of any r.e. class of infinite recursive sets.

Corollary 10. Let $C_0, C_1, \ldots$ be a recursive presentation of a class of infinite recursive sets. Let $B \notin P$ be recursive. Then there exists a recursive set A satisfying

1. $A \notin P$
2. $A \leq_p B$
3. $B \not\leq_{pT} A$
4. $C_i \not\subseteq A$ for any i.

Hence, the set A of Theorem 8 can be chosen so as not to have an infinite regular or context free subset.

By Theorem 1, the introduction of sufficiently large gaps into a set A results in a set A' satisfying $A' <_p A$. Together with a corollary to the following lemma this yields a strong version of Theorem 8.

Lemma 11. For any recursive A,B, if $A \cap \overline{B} \in P$, then $A \equiv_p A \cap B$.

Corollary 12. For any recursive B and NP-complete A, if $A \cap B$ is not NP-complete, then $A \cap \overline{B} \notin P$.

Corollary 13. Any NP-complete set A can be decomposed into sets B,C such that

1. $A = B \cup C$
2. $B, C \in NP-P$
3. B,C are not NP-complete

A similar result can be proved for other complete degrees (e.g., PSPACE).

Two recursive sets A_1 and A_2 form a minimal pair for P if neither is in P but $D \leq_{pT} A_0$ and $D \leq_{pT} A_1$ implies $D \in P$ for all D. Ladner [9] proved the existence of minimal pairs for P but did not give an upper bound of the complexity of the pair. Machtey[14] proved that subexponential minimal pairs exist. We are able to use the methods of Section 3 to obtain minimal pairs of non NP-complete sets in NP.

Theorem 14. Let $B \notin P$ be recursive. Then there are recursive A_0 and A_1 such that

a. $A_i <_p B$ $i = 0,1$
b. $A_i \notin P$ $i = 0,1$
c. $D \leq_{pT} A_i$ $i = 0,1$ implies $D \in P$.

Corollary 15. There exist $A, B \in NP-P$ which are not NP-complete such that $D \leq_p A$ and $D \leq_p B$ implies $D \in P$.

Corollary 16. For any $B \in NP-P$ there exists a minimal pair $A_0, A_1 \notin P$ such that $A_i <_p B$ $i = 0,1$.

By similar methods to the above we may prove a strong minimal pair theorem.

Theorem 17. Let $A, B \in P$ and $A <_p B$. Then there are C_1 and C_2 such that

a. $A <_p C_i <_p B$ $i=1,2$
b. $D <_p C_i$ $i=1,2$ implies $D \leq_p A$

Lynch [11], Berman [1] and Hartmanis and Berman [7] have studied the density of sets with respect to a number of different definitions. We can use Theorem 1 to obtain a result due to Lynch which states that if a set is reducible to arbitrarily sparse sets , then the set is in P.

5. ENUMERATION PROPERTIES

If a class of sets is recursively presentable, then there is an effective enumeration of devices which recognize all and only sets in the class[10]. The existence of such an enumeration is useful when studying properties of the class. In this section, we show that various important complete complexity classes are recursively presentable. Furthermore, the presentations can be in terms of efficient devices. Whereas the NP-complete sets are recursively presentable, this is not true of either NP-P or the non complete sets in NP-P. This may be interpreted as suggesting why so few intuitively appealing problems in NP ,which are not complete, have been found, i.e., perhaps most 'natural' problems are complete.

Theorem 18. If P=NP the class of NP-complete sets is recursively presentable. The presentation is an enumeration of algorithms which run in non-deterministic polynomial time and which accept all and only NP-complete sets.

The method of proof of the previous theorem can be used to obtain similar results for the complete sets of any time or space complexity class (PSPACE,EXP TIME,etc.) admitting an effective enumeration of the sets of the class by efficient devices (I.e., by devices whose complexity is bounded by the functions which define the class.)

In contrast to the above result, we can show that neither NP-P nor the non complete sets in NP-P are recursively presentable even if the only requirement on the devices is that they always halt, i.e., that they compute the characteristic functions of the sets considered. For NP-complete sets, the enumeration was in terms of non-deterministic polynomial time devices so that an enumeration exists in terms of deterministic exponential time devices for the characteristic functions.

It is important to notice that NP-P can not be recursively presented by characteristic function even if inefficient devices (e.g., non-primitive recursive) are permited.

We can also show that the recursively enumerable complete sets are not recursively presentable. While not surprising, this result does provide an indication that care should be exercised when considering definitional analogies between NP and the r.e. sets. The proof uses a recursion theoretic hierarchy argument.

REFERENCES

1. Berman, L. On the structure of complete sets: almost everywhere complexity and infinitely often speedup. Proc. Seventeenth IEEE Symp. on the Found. of Comp. Sci., 1975, 76-80.

2. Berman, P., Relationship between density and deterministic complexity of NP-complete languages. Fifth Int. Colloq. on Automata Languages and Programming, 1978,63-71.

3. Brainerd, W.S. and Landweber, L. H. Theory of Computation. Wiley, New York, 1974.

4. Cook, S.A. The complexity of theorem proving procedures. Proc. Third Annual ACM Symp. on Theory of Computing, 1971, 151-158.

5. Fortune, S. A note on sparse complete sets.

6. Meyer, A. R. and Paterson, M.S. With what frequency are intractable problems difficult? Laboratory for Computer Science Tech. Rep. TM-126, MIT, February 1979.

7. Hartmanis, J. and Berman, L. On isomorphisms and density of NP and other complete sets. SIAM J. Comp. 6(1977), 305-322.

8. Karp, R. M. Reducibility among combinatorial problems. In Complexity of Computer Computations, R. E. Miller and J. M. Thatcher, Eds., Plenum, New York,1975,85-103.

9. Ladner, R. On the structure of polynomial time reducibility. JACM 22(1975),155-171.

10. Landweber,L.H. and Robertson, E.L. Recursive properties of abstract complexity classes. JACM 19(1972),296-308.

11. Lynch,N. On reducibility to complex or sparse sets. JACM 22(1975), 341-345.

12. Machtey,M. The honest subrecursive classes are a lattice. Infor. and Contr. 24(1974) 247-263.

13. Rogers Jr.,H. Theory of Recursive Functions and Effective Computability. McGraw-Hill,NewYork,1967.

14. Machtey,M. Minimal pairs of polynomial degrees with subexponential complexity. TCS

15. Breidbart,S. The Structure of complexity classes and degrees. Technical Report, U. of CA.-Santa Barbara,1978.

16. Landweber,L.H.,Lipton,R.J. and Robertson,E.L. On the structure of sets in NP and other complexity classes. Technical Report 342,Comp.Sci. Dept.,U.ofWI, 1978.

AREA-TIME TRADEOFFS FOR MATRIX MULTIPLICATION AND RELATED PROBLEMS IN VLSI MODELS*

JOHN E. SAVAGE
Brown University, Computer Science Department, Providence, RI 02912

Abstract:

Two models for very-large scale integrated (VLSI) semiconductor circuits are considered that have been developed by Thompson and by Brent and Kung. The models permit the study of tradeoffs between chip area and computation time. We show that these tradeoffs can be derived from a single common complexity measure of a problem. We derive bounds on this measure for matrix multiplication under weak assumptions about the operations of addition and multiplication. The assumptions are such that the bounds can be applied directly to transitive closure and matrix inversion.

*This research was supported in part by NSF grants MCS 76-20023 and ENG 75-17614.

INTRODUCTION

Very large scale integration (VLSI) of semiconductor devices through the use of high resolution techniques has had a revolutionary effect on the computer industry. It has been responsible for the reduction of the cost of computing in the face of inflationary pressures in the marketplace. It has also changed the traditional measures of the cost of computers and the cost of computing. For example, the area occupied by a chip is now a better measure of cost than the number of discrete elements that it contains, although the latter is roughly proportional to the former. The new book by Mead and Conway [1] describes VLSI technology and provides an introduction to algorithms for the design of VLSI chips.

Thompson [2,3] has recently presented analysis of tradeoffs that are possible for VLSI chips between the chip area and the computation time of the chip. He demonstrates that $AT^2 = \Omega(n^2)$ for the discrete Fourier Transform on n inputs [2] and has similar results for other problems [3]. These results are derived by postulating a reasonable model for VLSI in terms of which these lower bounds can be derived. Brent and Kung [4] have presented a somewhat different model and have the same kind of result for the multiplication of two integers in the standard binary representation.

In this paper we consider the problem of matrix multiplication in both VLSI models and derive lower bounds of the form $AT^2 = \Omega(p^4)$ for multiplication of p x p matrices. We show that bounds for the two models can be derived from a common complexity measure. The conditions under which the bounds are derived are such that it also applies to matrix inversion and to transitive closure. The lower bounds for multiplication are achieved by algorithms developed by Kung and Leiserson [5] and by Preparata and Vuillemin [6].

VLSI MODELS

Thompson [2,3] and Brent and Kung [4] have presented models for VLSI circuits which have much in common. In this section we describe these models and summarize their accompanying analysis in terms that permit their application to arbitrary multi-output problems. A problem is characterized by a function $f:X^n \to X^m$, X a finite set, where $f(a_1, a_2, \cdots, a_n) = (b_1, b_2, \cdots, b_m)$ and a_i and b_j are input and output variables, respectively.

Both models assume that VLSI chips have nodes and wires with wires of width λ (determined by the resolution of photographic techniques) and spacing λ. The nodes serve as input ports, output ports, processing elements or a combination of several of these. Each node has an area of at least $\rho \geq \lambda^2$. The chip is laid out on several different planes, each of which contains wires and nodes and between which there are connections. The number of planes, $\gamma \geq 2$, is small.

The two models also assume that wires carry results from the alphabet X and that it takes one unit of time τ to transmit one result over one wire. In one of his models, Thompson [2] allows X to be nonbinary. Brent and Kung [4] assume that $|X| = 2$. The analysis of both models can be done assuming that X is nonbinary by increasing the wire width and spacing and by assuming a chip design in which wires are not split by selecting components of words in X.

The assumptions that are special to the Thompson (T) model [2] are the following:

T1. Wires are placed on a rectangular grid.

T2. Nodes with total degree d are square with sides of length $2\lambda d$.

T3. Each output variable appears in a distinct output port (or processing element).

The assumption special to the Brent and Kung (BK) model [3] is the following:

BK1. All computations are performed in a region R of the chip of area A, the outer boundary of which is convex.

Because assumption T3 is not made by Brent and Kung, they explicitly assume that more than one output variable may pass through an output port. Thompson could have made such an assumption without substantially weakening his results. Assumption T2 has been relaxed recently by Thompson [3] by requiring that the degree d of a node be limited to a small constant, say d = 4, which is reasonable, in which case, assuming a square node is a very weak condition. This restriction has little effect on the lower bounds. Thus, we conclude that the principal difference in the models is reflected in assumptions T1 and BK1 concerning the geometry of the chip.

We now turn to the analysis that has been done in these two models. For pedagogical reasons it is desirable to begin with the BK model, although it was developed after the T model. Using well-known relations between the area, perimeter and diameter of a convex region, Brent and Kung have shown that if C is the length of any chord perpendicular to a diameter of the convex region R of area A then

$$A \geq \frac{C^2}{\pi} \quad \cdot \quad (1)$$

In turn, if ω is the number of wires crossing this line

$$C \geq \omega\left(\frac{2\lambda}{\gamma}\right) \quad (2)$$

since each wire and spacing has width λ and at most γ wires overlap. Brent and Kung let M be the maximum number of the m output variables that are generated through any one port. They then show that by sliding a line perpendicular to a diameter, the output nodes can be divided into two disjoint sets, which correspond to a partition S_1, S_2 of the m output variables so that $|S_i|$ satisfies,

$$\frac{m-M}{2} \leq |S_i| \leq \frac{m+M}{2} .$$

Such a line partitions the inputs of the function $f:X^n \rightarrow X^m$, according to whether the input nodes that receive them fall on the same side as S1 or not. The amount of information which must flow across this line determines the area and time required by the chip and motivates the following new definition.

Definition Given $f:X^n \rightarrow X^m$, $f(a_1, a_2, \cdots, a_n) = (b_1, b_2, \cdots, b_m)$, let U_1, U_2 be a partition of $\{1, 2, \cdots, n\}$ and V_1, V_2 be a partition of $\{1, 2, \cdots, m\}$. For $V_s = \{j_1, j_2, \cdots, j_\ell\}$, let $f_{V_s} = (b_{j_1}, b_{j_2}, \cdots, b_{j_\ell})$ and let $f_{V_s}\Big|\begin{smallmatrix} a_{i_t} = c_t \\ i_t \notin U_r \end{smallmatrix}$ be the subfunction of f_{V_s} obtained by fixing the variables outside of U_r. Let $\mathscr{R}(g)$ denote the number of points which are the image under g of points in domain of g. Then, we define

$$\phi(V_r,U_s) = \left\lceil \log_X \max_{\{c_t \varepsilon X\}} \mathscr{R}\left(f_{V_r}\Big|\begin{matrix} a_{i_t} = c_t \\ i_t \notin U_s \end{matrix}\right)\right\rceil$$

which is a measure of the amount of information which may flow from input variables in U_s to output variables in V_r. The minimal crossflow I(M) is defined by

$$I(M) = \min_{\substack{U_1,U_2 \\ V_1,V_2 \ni \\ \left\lceil \frac{m-M}{2} \right\rceil \leq |V_i| \leq \left\lfloor \frac{m+M}{2} \right\rfloor}} \max\left(\phi(V_1,U_2),\ \phi(V_2,U_1)\right)$$

and is a measure of the minimal amount of information that may flow across any boundary that places inputs in U_1 and outputs in V_1 on one side of the boundary and the others on the other side under the constraint shown on $|V_1|$ and $|V_2|$.

It follows from the above definition that at least I(M) elements from X must be communicated across any chord of the convex region R. If the chord has ω wires, the time T required must satisfy

$$T \geq \frac{I(M)}{\omega} \tau \tag{3}$$

since transmission of each element requires τ seconds of which at most ω occur simultaneously. Combining equations (1), (2) and (3) we have

$$AT^2 \geq \frac{4}{\pi}\left(\frac{\lambda\tau}{\gamma}\right)^2 I^2(M) \tag{4}$$

This inequality is weak when M is large, since I(M) is then small, so we observe that at least $M\tau$ seconds are required to generate the M outputs produced through some port whose area $\rho \geq \lambda^2$. Thus,

$$AT^2 \geq (\lambda\tau)^2 M^2 \tag{5}$$

Combining equations (4) and (5) we have the desired generalization of the Brent-Kung theorem.

Theorem 1 In the Brent-Kung VLSI model [3] the area A and time T used by a VLSI chip to compute a function $f:X^n \rightarrow X^m$ satisfies the following inequality:

$$AT^2 \geq (\lambda\tau)^2 \max_{1 \leq M \leq m} \left[\frac{4}{\pi\gamma^2} I^2(M),\ M^2\right]$$

Thompson [2] assumes that the output variables appear in distinct ports or that M = 1. He defines ω, the minimal bisection width of a chip as the minimal number of wires that are cut by any line on the chip that divides the output ports into two disjoint sets $V_1{}^*$, $V_2{}^*$, whose size is as nearly equal as possible. It is clear that

$$\left\lceil \frac{m-1}{2} \right\rceil \leq |V_i{}^*| \leq \left\lfloor \frac{m+1}{2} \right\rfloor$$

He then demonstrates that the area A of the chip satisfies

$$A \geq \frac{\omega^2}{4} (2\lambda)^2$$

and shows that $T \geq I(1)\ \tau/\omega$ since only ω wires are available to carry the $I(1)$ units of information.

Theorem 2 In the Thompson VLSI model [2], the area A and time T used by a VLSI chip to compute a function $f:X^n \rightarrow X^m$ satisfies the following inequality

$$AT^2 \geq (\lambda\tau)^2\ I^2(1)$$

If Thompson's assumption T3 were relaxed, Theorem 2 would be identical with Theorem 1 except for constant multipliers. We show in the next section that both bounds are identical up to constant multipliers for matrix multiplication.

MULTIPLICATION OF SQUARE MATRICES

Consider the multiplication of two p x p matrices, A, B, over the set X where C = AB. Let the operations of addition and multiplication and the set X satisfy the following weak conditions:

M1. The set X is closed under the two operations.

M2. Additive and multiplicative identities 0 and 1 exist and for $x \varepsilon X$

$$0 \cdot x = 0$$

Since the object is to lower bound the minimal cross-flow, I(M), for matrix multiplication, we will show that a good lower bound can be achieved by setting elements in A or B (but not both) to form a permutation matrix, thus, these conditions on addition and multiplications will be sufficient.

Denote with two p x p binary matrices D_1 and D_2 a partition of the elements of a matrix D into two disjoint sets, where each non-zero entry in D_i identifies an element in the i th set. If D_i and E_i are two such p x p binary matrices, let $D_i \cap E_i$ denote their intersection which is the p x p matrix that has a non-zero entry exactly where both D_i and E_i have non-zero entries. Also, let $|D_i|$ be the number of non-zero entries in D_i. It follows immediately that $|D_1 \cap D_2| = 0$. We are now prepared to tackle matrix multiplication.

Let C_1 and C_2 denote a partition of C such that

$$\left\lceil \frac{p^2-M}{2} \right\rceil \leq |C_i| \leq \left\lfloor \frac{p^2+M}{2} \right\rfloor$$

and let A_1, A_2 and B_1, B_2 denote partitions of A and B. We shall show that many of the input variables in A_1 or B_1 can be mapped onto (or equated with) output variables in C_2 or that the same can be done for A_2, B_2 and C_1. An essential step in this demonstration will be to form matrices $A_1(i)$ and $A_2(i)$, which are cyclic column shifts of A_1 and A_2 for some $1 \leq i \leq p$ obtained by setting B to a cyclic permutation matrix, and to form matrices $B_1(j)$ and $B_2(j)$, $1 \leq j \leq p$, which are cyclic row shifts of B_1 and B_2 obtained by setting A to a cyclic permutation matrix.

Lemma 1 There exist integers $1 \leq i, j \leq p$, such that

$$|A_r(i) \cap B_r(j)| \leq \frac{|A_r|\ |B_r|}{p^2}$$

for r = 1,2.

Proof Consider the following sum:

$$S = \sum_{i=1}^{p} \sum_{j=1}^{p} |A_r(i) \cap B_r(j)|$$

Since $A_r(i)$ and $B_r(j)$ are formed by cyclic shifts of the columns of A_r and rows of B_r, respectively, each of the r_u 1's in the u th row of A_r encounters each of s_v 1's in the v th row of B_r for each $1 \leq v \leq p$. It follows that

$$S = |A_r|\ |B_r|$$

from which the hypothesis of the lemma follows directly. $\square$

Proposition 1 The minimal cross-flow I(M) for the multiplication of two p x p matrices over the finite set X is bounded by

$$p^2(.05) - \frac{M}{2} \leq I(M) \leq \frac{p^2 + M}{2}$$

Proof The upper bound is a direct consequence in the definition of I(M) and of the fact that $\mathscr{R}\left(f_{V_s} \middle| \begin{matrix} a_{i_t} = c_t \\ t \notin U_r \end{matrix}\right) \leq X^{|V_s|}$.

The lower bound proceeds from a series of observations. Suppose that for $0 \leq \varepsilon \leq 1$, $|A_r(i)| \geq \frac{p^2}{2}(1+\varepsilon)$ or $|B_r(j)| \geq \frac{p^2}{2}(1+\varepsilon)$ for some $r \varepsilon \{1,2\}$ and some value of i or j, $1 \leq i, j \leq p$. Then, for $r' \varepsilon \{1,2\}$, $r' \neq r$, either

$$|A_r(i) \cap C_{r'}| \geq \alpha\ (p^2,\varepsilon,M) \text{ or } |B_r(j) \cap C_{r'}| \geq \alpha\ (p^2,\varepsilon,M)$$

where

$$\alpha\ (p^2,\varepsilon,M) = (p^2\varepsilon - M)/2$$

when $M \leq p^2\varepsilon$ because

$$|D \cap C_{r'}| = |D| + |C_{r'}| - |D \cup C_{r'}| \geq |D| + |C_{r'}| - p^2$$

Thus we assume that

$$\frac{p^2(1-\varepsilon)}{2} \leq |A_r(i)|,\ |B_r(j)| \leq \frac{p^2(1+\varepsilon)}{2}$$

and derive a lower bound under these conditions. We then take the smaller of these two bounds. Observe that

$$\begin{aligned} &|C_{r'} \cap A_r(i)| + |C_{r'} \cap B_r(j)| \\ &\quad \geq |C_{r'} \cap (A_r(i) \cup B_r(j))| \\ &\quad \geq |A_r(i) \cup B_r(j)| + |C_{r'}| - p^2 \\ &\quad \geq |A_r(i) \cup B_r(j)| - \frac{p^2 + M}{2} \end{aligned}$$

so that if the union is large, then either $|C_{r'} \cap A_r(i)|$ or $|C_{r'} \cap B_r(j)|$ will be large and at least half the value of the lower bound. However, $|A_r(i) \cup B_r(j)| = |A_r(i)| + |B_r(j)| - |A_r(i) \cap B_r(j)|$. As a consequence we invoke Lemma 1 to show that there exist integers i, j such that

$$\begin{aligned} &|C_{r'} \cap A_r(i)| + |C_{r'} \cap B_r(j)| \\ &\quad \geq |A_r(i)| + |B_r(j)| - \frac{|A_r(i)|\ |B_r(j)|}{p^2} - \frac{p^2 + M}{2} \\ &\quad \geq p^2 \left[\frac{1}{2} - \varepsilon - \frac{(1+\varepsilon)^2}{4}\right] - \frac{M}{2} = \end{aligned}$$

Then, either

$$|A_r(i) \cap C_{r'}| \geq \beta(p^2,\varepsilon,M) \text{ or } |B_r(j) \cap C_{r'}| \geq \beta(p^2, \ ,M)$$

where

$$\beta(p^2,\varepsilon,M) = p^2\ \frac{1}{4} - \frac{\varepsilon}{2} - \frac{(1+\varepsilon)^2}{8} - \frac{M}{4}$$

when

$$\beta(p^2,\varepsilon,M) \geq 0.$$

The smaller of $\alpha(p^2,\varepsilon,M)$ and $\beta(p^2,\varepsilon,M)$ is a lower bound on $I(M)$ because at least this many variables can be mapped onto C_1 or C_2 by choosing A or B to be an appropriate permutation matrix. Here $0 \leq \varepsilon \leq 1$ is arbitrary, so we may choose it to make the smaller of α and β as large as possible. It is sufficient to choose ε such that

$$\frac{\varepsilon}{2} = \frac{1}{4} - \frac{\varepsilon}{2} - \frac{(1 + \varepsilon)^2}{8}$$

or $\varepsilon = .099$ in which case α is the smaller of the two. Then,

$$I(M) \geq \alpha(p^2,\ .099,\ M) = p^2(.05) - \frac{M}{2}$$

This bound on the minimal cross-flow is now applied to lower bound the VLSI complexity measure AT^2.

Theorem 3 Let $C = AB$ denote multiplication of p x p matrices over a set X where X and the operations of addition and multiplication satisfy conditions M1 and M2. Then the area A and time T in the Brent-Kung (BK) and Thompson (T) VLSI models to compute C must satisfy the following inequalities:

$$AT^2 \geq (\lambda\tau)^2(p^2(.05) - .5)^2 = \Omega(p^4) \qquad (T)$$

$$AT^2 \geq (\lambda\tau)^2 \left[\frac{p^2(.05)}{1 + \sqrt{\pi\gamma}}\right]^2 = \Omega(p^4) \qquad (BK)$$

Proof In the BK model the lower bound requires the computation of $\max\left[\frac{4}{\pi\gamma^2}\ I^2(M),\ M^2\right]$, which due to the monotonicity of $I(M)$ is achieved when $M = I(M)2/(\gamma\sqrt{\pi})$

Kung and Leiserson [5] have designed a hex-connected cellular sequential machine which multiplies two p x p matrices of bandwiths ω_1 and ω_2 in $3p + \min(\omega_1,\omega_2)$ units of time. The machine consists of a number of small identical cells and has an area proportional to $\omega_1\ \omega_2$. Since the matrices that we consider are full, $\omega_1 = \omega_2 = 2p - 1$, and $A = \Theta(p^2)$, $T = \Theta(p)$ so $AT^2 = \Theta(p^4)$, which meets our lower bounds up to a multiplicative factor. Preparata and Vuillemin [6] have designed a machine consisting of cube-connected cycles of processors, which can also perform matrix multiplication with $A = \Theta(p^2)$ and $T = \Theta(p)$.

REMARKS AND CONCLUSIONS

Two models for VLSI circuits have been considered, one which assumes that wires are laid out on a rectangular grid, and another which assumes that the circuit occupies several superimposed planes, the common outer boundary of which defines a convex planar region. For each model we have shown that multiplication of square p x p matrices requires a product of area A and computation time T which satisfies an inequality of the form $AT^2 = \Omega(p^4)$.

These results are derived under weak conditions on the two operations of addition and multiplication and the set X over which they are defined. The conditions are such that they apply to semirings for which it has been shown [7] that the transitive closure of a 3p x 3p matrix reduces to matrix multiplication of p x p matrices. Also, the conditions are such that the inverse of a 3p x 3p matrix over a field reduces to matrix multiplication of p x p matrices. We conclude that any VLSI circuit for either transitive closure or matrix inversion must also satisfy the inequality $AT^2 = \Omega(p^4)$. A VLSI circuit for transitive closure has been given by Guibas et al [8] which achieves the lower bound with $T = O(p)$.

Lower bounds have also been derived for the multiplication of non-square matrices. By a somewhat more involved argument it can be shown that the product AT^2 is no larger that $\Omega(m^2p^2)$ for the multiplication of m x n and n x p matrices. It is not known how tight this result is when m, n and p are not comparable.

REFERENCES

1. Mead, C. A. and L. A. Conway, Introduction to VLSI Systems, Addison-Wesley, Menlo Park, CA, 1979.

2. Thompson, C. D., "Area-Time Complexity for VLSI, "Proc. 11th Annual ACM Symposium on Theory of Computing," ACM, New York, pp 81-88, 1979.

3. Thompson, C. D., "A Complexity Theory for VLSI," Ph.D. Thesis, Department of Computer Science, Carnegie-Mellon University, 1979.

4. Brent, R. P. and H. T. Kung, "The Area-Time Complexity of Binary Multiplication," Technical Report CMU-CS-79-05, Department of Computer Science, Carnegie-Mellon University, 1979.

5. Kung, H. T. and C. E. Leiserson, "Systolic Arrays for VLSI," in [1] and Technical Report, Department of Computer Science, Carnegie-Mellon University, 1979.

6. Preparata, F. P. and J. Vuillemin, "The Cube-Connected-Cycles: A Versatile Network," Proc. 20th Annual Symposium on Foundations of Computer Science, IEEE Computer Society, 1979.

7. Aho, A., J. Hopcroft and J. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley, Menlo Park, CA, p. 203, 1974.

8. Guibas, L. J., H. T. Kung and C. D. Thompson, "Direct VLSI Implementation of Combinatorial Algorithms," Technical Report, Department of Computer Science, Carnegie-Mellon University, March, 1979.

SOME SPACE-EFFICIENT ALGORITHMS - extended abstract

JOSEPH JA'JA' AND JANOS SIMON
Department of Computer Science, The Pennsylvania State University
University Park, PA 16802

0. Abstract. We present several new algorithms for some well-known problems, like matrix inversion, planarity testing, computing the transitive closure of a relation, finding a minimum-cost spanning tree and determining the biconnected and triconnected components of a graph. The algorithms require only $O(\log^2 n)$ bits of storage, where n is the size of the input, while the previously known algorithms use $O(n)$ bits.

1. Introduction. One of the major open problems in computational complexity is understanding the relationships between time and space complexities. While it is known that space is more powerful than time [10], it is not known whether there is an exponential gap between the space and time requirements of all algorithms for some problem. It is also conjectured that P contains languages that are not recognizable in $\mathrm{DSPACE}[\log n^i]$ for any i but again, no proofs exist [3]. (We denote by $\mathrm{DSPACE}[f(n)]$ ($\mathrm{DTIME}[f(n)]$) the class of languages recognized by deterministic Turing machines with workspace limited to $f(n)$ squares of tape for inputs of length n (with $f(n)$ moves, respectively). It is known that the space measure is quite invariant with respect to the machine model, so that $\mathrm{DSPACE}[f(n)]$ might have been defined as the class of languages recognized by random access machines with a read-only input register that use $O(f(n))$ bits to recognize inputs of length n. This will be our model of computation--the instruction set will consist of the four arithmetic operations on positive integers and comparison-based branches.)

In this paper we show that many interesting problems, known to be in P have space-efficient algorithms, by exhibiting algorithms that use $O(\log^2 n)$ bits of storage. The saving of memory comes from extensive recomputing of previous results, together with careful bookkeeping strategies, and, in some cases, from a reformulation of the problem that enables us to use such algorithms. We consider these results interesting for several reasons. They refine our intuition about what can be done in very little space--in particular they seem to indicate that the class $\mathrm{DSPACE}[\log^2 n]$ is worth much further study. The results show that much less information must be recorded during some computations than one would naively believe. The two natural questions--can the $\log^2 n$ bound be improved? and can these techniques be made to work for problems like matchings and flows in graphs, where no space-efficient algorithms are known--seem very challenging. Finally it should be noted that these results have obvious connections to parallel computations [5], [2]--an issue we explore in an independent paper [7].

2. Matrix Algorithms. We start by presenting an easy algorithm for addition of binary numbers. Although the algorithm is straightforward, it is a basic building block in more sophisticated procedures. It also allows us to present the formalism.

There is a basic problem in space-efficient algorithms where the workspace is smaller than the answer--how is the answer given? For example, two n-digit binary numbers can be added in $O(\log n)$ workspace, but the result has $O(n)$ bits. The usual formalism is to define a write-only output medium. This has the disadvantage that function composition is not possible. We shall say that $f(x)$, is computable in space $s(x)$ if the function

$$\hat{f}(x, i) = \begin{cases} \text{the } i\text{-th bit of } f(x)\text{, if } |f(x)| \geq i \\ \$ \quad \text{ow} \end{cases}$$

is computable in space $s(x)$. We shall use functions like $\hat{f}$ throughout this section, to build up our matrix power and matrix inversion algorithms.

Suppose that k integers $X_1, X_2, \ldots, X_k$ are given as a function $\hat{X}(i, b) =$ the b-th bit of X_i. We present an algorithm, Add, to compute $\sum_{i=1}^{n} X_i$ --i.e. Add(m) will return the m-th bit of the sum.

We assume, for simplicity, that the internal representation of integers uses \$ for leading 0's; \$ behaves as 0 in arithmetic operations. This enables us to talk of the i-th bit of an integer that has fewer than i bits. The least significant bit is in position 0. Thus, for example, the number 5 (101) has 1 as its 0-th and second bits, 0 as its first bit and \$ as its third, fourth, etc. bits. Clearly, the use of \$ could have been avoided, by using more complicated programs

```
Procedure Add(X, k, m)  /*returns the m-th bit of the*/
   CARRY ← 0            /*sum $\sum_{i=1}^{k} X_i$               */
   do COLUMN = 0 to m
      SUM ← 0
      do i = 1 to k /*compute sum of bits in this column*/
         SUM ← SUM + X(i, COLUMN)
      end
      SUM ← SUM + CARRY
      CARRY ← ⌊SUM/2⌋  /*this is the new carry*/
   end
   return(last-bit-of (SUM))
end Add
```

The correctness of Add is straightforward. The functions ⌊ ⌋ (floor), and last-bit-of can be easily implemented in our RAM model. Similarly, the do loops are simply a shorthand.

The space requirements are the following:

$\lceil \log_2 k \rceil$ bits are necessary for i

$\lceil \log_2 m \rceil$ bits are necessary for COLUMN

$\lceil \log_2 k \rceil$ bits may be necessary for SUM and CARRY.

In addition, space must be reserved for a single invocation of X. Since COLUMN (and m) and k are used as inputs for X, the procedure uses space proportional to the space needed for X, and thus, it is optimal.

We present now an algorithm to multiply two integers. It simulates the standard binary multiplication, doing shifts and additions--except nothing is actually written down. Bits are recomputed as they are needed. As in Add, computation proceeds from left to right, to compute carries, (see [5] for a similar algorithm).

```
Procedure Mult(X, Y, m)  /*returns the  m-th  bit of the product  XY*/
   CARRY ← 0
   do  COLUMN = 0  to  m
      XPTR ← COLUMN; SUM ← 0
      do  YPTR = 0  to COLUMN  /*loop invariant:  XPTR + YPTR = COLUMN*/
                               /*SUM contains all contributions to*/
                               /*product from first  XPTR  bits of  X*/
          SUM ← SUM + X(XPTR) * Y(YPTR)
          XPTR ← XPTR - 1
      end
      SUM ← SUM + CARRY
      CARRY ← SUM/2
   end
   return last-bit-of (SUM)
end Mult
```

The space requirements of Mult are as follows: since all index variables are bounded by m, a parameter in the call for X and Y, within a constant factor, these space requirements are those of X. The maximum value of SUM is $2 * \text{length}(Y) - 1$, and so a buffer of length $O(\log(Y))$ suffices. The total space requirements are $O(\log_2 Y + \text{space}(X) + \text{space}(Y))$ where space(U) is the amount of space needed to obtain a bit of U. In particular if X and Y are in a read-only memory, Mult uses $O(\log n)$ space.

Using the routines above, it is not hard to obtain a space-efficient matrix multiplication algorithm. Given two $n \times n$ matrices A and B (i.e. functions $A(i, j, m)$, $B(i, j, m)$, that return the m-th bit of element A_{ij} (B_{ij})), the element C_{ij} of the product matrix $C = AB$ is given as $\text{Add}(U_{ij}, n, m)$ where $U_{ij}(k) = A_{ik} * B_{kj}$ (i.e. $\text{Mult}(X_{ik}, Y_{kj}, m)$). The procedure can still be carried out in $O(\log n)$ space (plus the space needed for obtaining the operands).

We will consider now the problem of computing powers of matrices. We define exponentiation by induction on the length of the exponent as

$$A^0 = 1$$

$$A^{x0} = A^x \cdot A^x$$

$$A^{x1} = (A^x \cdot A^x) \cdot A.$$

If A is $n \times n$ and the length of the elements A_{ij} in a binary representation is at most $O(n)$ then the length of entries $A^x_{\ell m}$ of A^x is at most $O(xn)$. In particular, if $x \leq n$, $O(\log n)$ memory is sufficient to address any bit of $A^x_{\ell m}$.

To obtain A^x, let $x = ya$ ($a = 0$ or 1), and suppose, by induction, that a routine $A_y(i, j, m)$ can produce the m-th bit of element A^y_{ij} of A^y, using space $s(|y|)$. Then our routines, with the recurrence relations, can produce any bit of any element of A^x, in space $s(|y|) + O(\log n)$. If $x \leq n$, this yields an $O(\log^2 n)$ procedure, assuming A_{ij} has $O(n)$ bits.

Finally, we observe that the same techniques are applicable to matrix inversion. Recall that if A is an invertible $n \times n$ matrix then

$$A^{-1} = -\frac{A^{n-1} + c_1 A^{n-2} + \ldots + c_{n-2}A + c_{n-1}I}{c_n}$$

where the c_i's are the coefficients of the characteristic polynomial of A. The coefficients c_i are the solution of the triangular linear system $Sc = s$

$$\begin{bmatrix} 1 & & & & & \\ s_1 & 2 & & & 0 & \\ s_2 & s_1 & 3 & & & \\ s_3 & s_2 & s_1 & 4 & & \\ \vdots & \vdots & \vdots & & \ddots & \\ s_{n-2} & s_{n-3} & \vdots & & & \\ s_{n-1} & s_{n-2} & s_{n-3} & \cdots & s_1 & n \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ \vdots \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ \vdots \\ \vdots \\ s_n \end{bmatrix}$$

with $s_i = \text{trace}(A^i)$. [4]

We can compute the matrix S and the vector s using $O(\log^2 n)$ space. Assuming that the c_i's may be obtained in $O(\log^2 n)$ space, we could obtain A^{-1}.

To solve a triangular system use the recurrence [6] if

$$B = \begin{bmatrix} B_{11} & 0 \\ B_{21} & B_{22} \end{bmatrix} \quad \text{with } B_{11}, B_{22} \text{ triangular}$$

$$B^{-1} = \begin{bmatrix} B_{11}^{-1} & 0 \\ -B_{22}^{-1}B_{21}B_{11}^{-1} & B_{22}^{-1} \end{bmatrix}.$$

Again, the depth of recursion is $\log n$, with $O(\log n)$ space necessary in every invocation for bookkeeping. In addition, $O(\log^2 n)$ space is used to yield the necessary bits of the arguments, a total space requirement of $O(\log^2 n)$.

Our results are summarized in the following:

Theorem 1. Assume all integers in the input are at most $O(n)$ bits long. Then there are algorithms to compute the tasks below, with the indicated space bounds

a) Addition of two integers	$O(\log n)$
b) Product of two integers	$O(\log n)$
c) Product (or sum) of two $n \times n$ matrices	$O(\log n)$
d) k-th power of an $n \times n$ matrix	$O(\log k \log n)$
e) Computing the inverse of an invertible $n \times n$ matrix	$O(\log^2 n)$. ☐

Corollary. The following functions are computable in $\mathrm{DSPACE}[\log^2 n]$:

a) A^* where A is a relation over n objects and $*$ indicates reflexive and transitive closure

b) $\det(A)$ where A is an $n \times n$ matrix with n-bit entries, as well as solving linear systems having A as the coefficient matrix.

The proofs will be left to the full paper. ☐

Note that a) of the corollary above is a special case of a general result about complete semirings [1]. Whenever the size of the representation of the elements does not grow faster than in ordinary multiplication, our techniques are applicable. Thus, most problems that can be solved using a 'dynamic programming' approach have algorithms using $O(\log^2 n)$ space--for example, finding shortest paths in graphs (without negative cycles).

3. Some Applications. The main result of this section is that planarity testing can be done in $O(\log^2 n)$ space. In addition, if the graph is planar, we can obtain the cartesian coordinates of the vertices in a plane embedding.

The algorithm uses as subroutines space-efficient routines for finding spanning trees, biconnected and 3-connected components, as well as the matrix inversion routine of the preceding section. In this preliminary version, due to space limitations, we shall give only a rough outline of the method.

Recall that a graph is planar iff all its triconnected components are planar [9]. We shall develop an algorithm for testing planarity of triconnected planar graphs, together with an algorithm to decompose a graph into its 3-connected components, to test planarity of arbitrary graphs.

The planarity algorithm uses the ideas of Tutte's characterization of plane 3-connected graphs and their barycentric plane embeddings [12].

Definition. A cycle C in a graph G is peripheral if the graph induced by the set of edges of G that are not edges of C, is connected. Informally, the removal of C does not disconnect G.

The main steps in our algorithm are:

1) Find a peripheral cycle C of G. Let C have m vertices.

2) Embed C in the cartesian plane as a convex polygon. Let the coordinates of the vertices be (x_i, y_i) $1 \le i \le m$.

3) Let the matrix B be defined as

$$B_{ij} = \begin{cases} -1 & \text{if } i \neq j \text{ and there is an edge between vertex } i \\ & \text{and vertex } j \\ 0 & \text{if } i \neq j \text{ and } i \text{ and } j \text{ are not connected by an} \\ & \text{edge; } \underline{\text{degree}}(v_i) \text{ ow} \end{cases}$$

Solve the system of equations on the unknowns $x_{m+1}, x_{m+2}, \ldots, x_n, y_{m+2}, \ldots, y_n$

$$\sum_{j=1}^{n} B_{ij} x_j = 0$$

$$\sum_{j=1}^{n} B_{ij} y_j = 0.$$

4) The coordinates (x_i, y_i) $1 \le i \le n$ obtained from 2) and 3) are a plane embedding of G if G is planar and 3-connected. Verify that this is so, by checking that for every pair of vertices v_ℓ, v_m connected by an edge e, for every other pair of vertices v_s, v_t connected by an edge, with $\{v_\ell, v_m\} \cap \{v_s, v_t\} = \phi$, the segments (v_ℓ, v_m) and (v_s, v_t) do not have a common point.

Note that step 3 forces every vertex not on the original cycle to be located at the center of gravity of the vertices directly connected to it, assuming unit weights on all vertices.

To implement step 1, we use the facts that in nontrivial 3-connected graphs, the shortest cycle is always peripheral, and that planar 3-connected graphs have a shortest cycle with at most 5 edges. Thus, step 1 can be implemented by a search through all triples of vertices, that check whether the vertices form a triangle, followed in case of failure by a search through all 4-tuples, and, possibly through all quintuplets of vertices.

Since the peripheral cycles found by step 1 are so simple, step 2 is also easy to implement--we use some standard triangle, square or pentagon.

Steps 3) and 4) involve solving linear equations. We use the algorithms developed in the previous section.

As for space requirements, steps 1) and 2) require $O(\log n)$ bits, since we have to store only a fixed number of vertices at any time. Step 3 requires $O(\log^2 n)$ bits. While step 4) could be done in $O(\log n)$ bits, it calls on step 3) to recompute the coordinates of the vertices in the embedding. The total space complexity is $O(\log^2 n)$. □

We sketch our algorithm for obtaining the three-connected components (tcc's) of a graph $G = (V, E)$. We use McLane's definition of tcc [9]. The following facts will be proven in the full paper:

Fact 1: Vertices i and j of G are in the same tcc iff for all vertices $\alpha, \beta, \alpha, \beta \neq i, j$ the edge $(i, j) \varepsilon E^*_{\alpha\beta}$, where $G_{\alpha\beta} = (V_{\alpha\beta}, E_{\alpha\beta})$ is the subgraph of G induced by $V - \{\alpha, \beta\}$ and $G^*_{\alpha\beta} = (V_{\alpha\beta}, E^*_{\alpha\beta})$ is the transitive closure of $G_{\alpha\beta}$.

Let R be the relation on $V \times V$ defined by iRj iff i and j are in the same tcc.

Define the relation R' on $V \times V \times V$ by

$(i, j, k) \varepsilon R'$ iff $iRj \wedge iRk \wedge jRk$

For $(i, j, k) \varepsilon R'$, let

$V_{ijk} = \{\ell \varepsilon V | \ell Ri \wedge \ell Rj \wedge \ell Rk\}$.

Fact 2: Each V_{ijk} determines a tcc and each tcc can be determined in such a way.

Thus, to obtain the tcc's of G it suffices to compute R'. The space requirements to compute R' are $O(\log n)$ (to store i, j, k) plus the space required to compute R. Computing R involves, essentially, a transitive and reflexive closure, that can be done in $O(\log^2 n)$ space. Therefore, $O(\log^2 n)$ space suffices to obtain the tcc's of a biconnected graph.

A similar algorithm, with an $O(\log^2 n)$ space bound, yields the biconnected components of a graph. The algorithm for minimum cost spanning trees, an adaptation of Sollin's method, has the same asymptotic bound. Both of these will be presented only in the full paper. Related results appear in [11].

4. Conclusions. We obtained space-efficient algorithms for many problems, for which good (polynomial time) algorithms exist. These results, together with earlier work, showing that other interesting problems, such as context-free language recognition are in DSPACE$[\log^2 n]$ [8] suggest that the class DSPACE$[\log^2 n]$ deserves further study. Clearly, 'divide and conquer' strategies where stack depth $O(\log n)$ suffices (and a total depth $O(n^k)$ for some k needs to be explored), and where each invocation (stack frame) can be recorded as a finite number of addresses, will yield an $O(\log^2 n)$ space bounded algorithm. We obtained a number of such strategies--some of them are quite different from the usual algorithm that would be used to solve the problem. We believe that the added insight into the nature of DSPACE$[\log^2 n]$ that the existence of these algorithms provide is the main virtue of our results. In addition, parallel versions of these algorithms [7] may be actually useful. Many interesting (and apparently difficult) questions are suggested by these results. For example:

- which of these problems are solvable in $O(\log n)$ nondeterministic space?

- what kind of (lower and upper bounds on) space-time trade-offs can we exhibit?
- can the $\log^2 n$ bound be improved?
- for all the algorithms, parallel implementations with time bounds exist, with time bounds that exactly match the space bounds. Can we prove this in general?

Bibliography

[1] Aho, A., J. E. Hopcroft and J. Ullman: The design and analysis of computer algorithms Addison-Wesley, Reading, Mass. (1974).

[2] Chandra, A. K. and L. Stockmeyer: 'Alternation' Proc. 17th Ann. FOCS Symp. (1976) 98-108.

[3] Cook, S. A: 'An observation on time-storage tradeoff' JCSS 9 (1974) 308-316.

[4] Csanky, L: 'Fast parallel matrix in version algorithms' SIAM J. Comp. 5 (1976) 618-623.

[5] Hartmanis, J. and J. Simon: 'On the power of multiplication in random access machines' Proc. 15th SWAT Symp. (1974) 13-23.

[6] Heller, D: 'A determinant theorem with applications to parallel algorithms' Tech. Rept., Dept. Comp. Sci. Carnegie-Mellon University (March 1973).

[7] Ja'Ja', J. and J. Simon: 'Parallel algorithms in graph theory planarity testing' Tech. Rept. Dept. of Comp. Sci., The Pennsylvania State University (to appear).

[8] Lewis, P. M., R. E. Stearns and J. Hartmanis: 'Memory bounds for recognition of context-free and context-sensitive languages' IEEE Conf. Rec. Switch. Th., Ann Arbor, Mich. (1965) 191-202.

[9] MacLane, S: 'A structural characterization of planar combinatorial graphs' Duke Math. J. 3 (1937) 460-472.

[10] Paul, W. J. and R. Rüdiger: On space vs. time II. Proc. 20th FOCS Conf. (1979).

[11] Savage, C. and J. Ja'Ja': 'Fast efficient parallel algorithms for some graph problems' submitted for publication.

[12] Tutte, W. T: 'How to draw a graph' Proc. London Math. Soc. B 13 (1963) 743-768.

This research was partially supported by National Science Foundation grants MCS 7827600 and MCS 7905006.

APPLICATION OF ESTIMATION TECHNIQUES TO TARGET TRACKING: AN OVERVIEW OF PROBLEMS AND SOLUTIONS*

C. B. CHANG AND J. A. TABACZYNSKI
Lincoln Laboratory, M.I.T.
Lexington, Massachusetts 02173

ABSTRACT

In this paper, we present a survey of problems and solutions in the area of target tracking. Discussions include design trade-offs and current issues.

1. INTRODUCTION

The target tracking trajectory estimation problem has been for nearly two decades a fruitful area of application for state estimation techniques. Many problems have been solved yet new and diversified applications still challenge systems engineers. In this paper, we provide a review of the subject and point out some current issues. The emphasis of this paper is not algorithmic detail, but rather the survey of problem areas and the discussion of various approaches. References containing specific estimation algorithms are cited for those readers interested in detail.

This paper is organized as follows. The fundamental problem of tracking is reviewed in Section 2. Along with this we discuss the basic solution approaches found in the current literature and describe some application. In Section 3 we present four approaches for tracking targets capable of executing sudden maneuvers. Discussions include a comparison of their relative merits. The problem of tracking with passive sensors where measurements contain only line-of-sight angles is discussed in Section 4. In certain applications, one is confronted with large scale system problems, e.g., the operation of multiple sensors in a dense multi-target environment. The paper concludes by discussing several approaches to processing multiple sensor data in Section 5 while Section 6 considers the problem of tracking in a multi-target environment.

2. FUNDAMENTALS

2.1. Problem Definition

The tracking problem is a state estimation problem, i.e., assuming the state vector of a target evolves in time according to the equation

$$\dot{x} = f(x) + w \tag{2.1}$$

and the corresponding (discrete) measurement vector** is given by

$$z_k = h(x_k) + v_k \tag{2.2}$$

where w and v_k being input and measurement vector noise processes, one is interested in estimating the target state vector x_k based upon the entire sequence of measurement vectors z_ℓ, $\ell=1,\ldots,k$.

We make these following remarks.

(1) Eq. (2.1) is a mathematical model representative of the target dynamics. The state vector x_k usually contains the target

---

*This work was sponsored by the Department of the Army. The views and conclusions contained in this document are those of the contractor and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the United States Government.

**The measurement device can be a radar, sonar, telescope (passive), and others. In most cases, these measurements are taken at discrete times.

position, velocity, and sometimes acceleration as state variables. In some situations, key parameters characterizing important target properties are also included as state variables.

(2) Eq. (2.2) is the measurement equation which relates state variables to measurement variables. When a radar is used, z_k has at least three components. i.e., range and two angles. If a passive sensor (such as a telescope) is used, z_k only contains two angle measurements.

(3) The system (input) and measurement noise processes w and v_k, respectively, may not be zero mean white noise processes. For example, a target maneuver may be modeled as an unknown mean vector of w_k. The covariance of w_k, Q, is selected to compensate for modeling errors (discrepencies between (2.1) and the true physical process).

2.2. Basic Approaches

The basic tracking filter is a recursive algorithm. Over the time interval t_k to t_{k+1}, the state estimate is propagated by

$$\dot{x} = f(x) \tag{2.3}$$

using the prior state vector estimate, $\hat{x}_{k/k}$ as the initial condition. At time t_{k+1}, a new measurement z_{k+1} is obtained, and the state estimate is updated by

$$\hat{x}_{k+1/k+1} = \hat{x}_{k+1/k} + K_{k+1}(z_{k+1} - h(\hat{x}_{k+1/k})) \tag{2.4}$$

where $\hat{x}_{i/j}$ is the estimate of x_i based upon all measurements z_k, k=1,...,j. Two questions arise:

(1) How does one choose f( )?

(2) How is the filter gain K_{k+1} computed?

These two questions which appear different initially are actually intimately related and are discussed below.

2.2.1. Target Dynamics

For most problems dealing with the tracking of aircraft, a constant velocity (CV) dynamic model is used, i.e., the state vector contains six variables

$$x = [x_1,x_2,x_3,x_4x_5,x_6]^T = [\zeta_1,\dot{\zeta}_1,\zeta_2,\dot{\zeta}_2,\zeta_3,\dot{\zeta}_3] \tag{2.5}$$

where ζ_i is the ith coordinate of the three dimensional position coordinate system chosen to represent the target dynamics. The state equations are

$$\begin{aligned} \dot{x}_i &= x_{i+1} \\ \dot{x}_{i+1} &= w_i \end{aligned} \tag{2.6}$$

for i=1,3,5, where w_i is a process noise term used to characterize modeling errors.

If the target being tracked maneuvers "slowly" a constant acceleration (CA) dynamic model is usually required, i.e.,

$$x = [\zeta_1,\dot{\zeta}_1,\ddot{\zeta}_1,\zeta_2,\dot{\zeta}_2,\ddot{\zeta}_2,\zeta_3,\dot{\zeta}_3,\ddot{\zeta}_3] \tag{2.7}$$

The state equations can be written accordingly. These two models are also referred to as the first and second order polynomial dynamics, respectively.

These models are often used for tracking aircraft. They may also be used for tracking ballistic targets. In some applications to ballistic targets, aerodynamic parameters of the vehicle such as the ballistic and lifting coefficients must be estimated in real time. This requires a set of nonlinear differential equations to describe the target dynamics. State models for ballistic targets are discussed in [1]-[3], [7].

2.2.2. Filter Gain Computation If an extended Kalman filter is used, the filter gain is computed using

$$K_k = P_{k/k} H_k^T R_k^{-1} \tag{2.8}$$

where H_k is the measurement Jacobian matrix, R_k is the measurement noise covariance, and $P_{k/k}$ is the covariance of the state estimate $\hat{x}_{k/k}$ obtained by solving the matrix Riccati equation.

(1) Over the interval t_k to t_{k+1}, compute $P_{k+1/k}$ by solving

$$\dot{P} = FP + PF^T + Q \quad , \quad P_{k/k} \text{ as initial condition} \tag{2.9}$$

(2) At t_{k+1}

$$P_{k+1/k+1} = P_{k+1/k}[I - H_{k+1}^T (H_{k+1} P_{k+1/k} H_{k+1}^T + R_{k+1})^{-1} H_{k+1} P_{k+1/k}] \tag{2.10}$$

where F is the system Jacobian matrix and Q is the process noise covariance matrix. The matrix Q is selected to be representative of modeling errors. For example, if a CV model is chosen to model a target which is actually accelerating, Q should be proportional to the expected magnitude of target acceleration. A method for selecting Q such that the covariance computed by the filter is an upper bound on the actual performance is suggested in [8]. Depending upon a specific application Q may also be determined empirically using Monte Carlo simulation [7].

There exist other methods for compensating modeling errors. These include the finite memory filter [1] and the fading memory filter [1], [9]. The finite memory filter employs a sliding data window and solves for the P matrix corresponding to the data window length. With the measurement noise covariance, data rate, and expected modeling error given, one is able to determine an optimum window width which minimizes the mean square error. The fading memory filter (sometimes referred to as the aging filter) weighs the recent data exponentially higher than the past data. A derivation of this filter is found in [9]. Similarly, with measurement noise covariance, data rate, and expected modeling error given, one is able to obtain an optimum weighting factor for minimizing the mean square error .

In some cases, because of computational constraints, it may be impractical to compute the filter gain in real time. In such event either a set of precomputed filter gains or a constant gain filter will have to be used. A commonly used constant gain filter employs the steady state solution of the Riccati equation, [1]. Another popular constant gain filter is the α-β-γ filter (or α-β filter in the case of a CV model) where values of α, β, and γ are filter gains for position, velocity, and acceleration estimates, respectively, [5]-[6]. The main difference between the α-β-γ filter and the general steady state gain filter is that the former assumes the complete independence of the three spatial coordinates in the filter update equation. Notice that the state space and the measurement space may be related through a nonlinear function (Eq. (1.2)). In using the α-β-γ filter, this assumption is no longer true. There is a wide range of methods for choosing the values of α, β, and γ. One method is to use the steady state Kalman gain. Other methods include extensive Monte Carlo simulation studies to define α, β, γ over a variety of cases in a given application.

3. TARGETS WITH SUDDEN MANEUVERS

Tracking filters discussed in the previous section can be applied to targets which maneuver "slowly". In this section, we discuss the problem of tracking targets which can perform sudden and large maneuvers. Targets having a capability to execute sudden maneuvers can be modeled as systems with abrupt changes. We modify equation (2.1) to become

$$\dot{x} = f(x,x_m) + w \tag{2.1a}$$

where x_m is a vector representing the maneuvering force.

In tracking aircraft for example, the target may first be flying in a straight line with constant speed. A CV model is an adequate model of aircraft dynamics in this case. A sudden maneuver of this aircraft implies that the plane is accelerating unexpectedly and the acceleration may be time-varying and could follow an unknown profile. The instantaneous acceleration vector is therefore the maneuvering vector x_m defined above.

We discuss four approaches to this problem below.

3.1. Filter Compensation Using Process Noise Covariance

In this first method, one simply ignores the maneuver vector x_m and lumps the system errors introduced by x_m with the process noise term w. If the only concern of the estimators is to maintain the target in track (adequate position estimation accuracy), this method can work quite well.

Basically, it examines the "regularity" of the filter residual vector γ_k against its covariance matrix P_{γ_k} using

$$\ell_k = \gamma_k^T P_{\gamma_k}^{-1} \gamma_k \tag{3.1}$$

When ℓ_k becomes too large, one suspects that the target is maneuvering, and responds by increasing Q, the covariance of w, so that ℓ_k is reduced to a reasonable value. This method has the combined feature of maneuver detection and filter compensation.

A thorough discussion of this method and its performance against maneuvering vehicles is given in [7]. This filter is based upon the adaptive filter of Jazwinski, [10].

3.2. State Augmentation

The second method is straightforward, but computationally more costly. However, it does provide substantially better performance than the previous method. This method is to include x_m as part of the state vector, i.e., the augmented state consists of

$$x_a = [x^T, x_m^T]^T \tag{3.2}$$

In the case where the maneuvering target dynamics is completely unknown, one assumes

$$\dot{x}_m = w_m \tag{3.3}$$

where w_m is modeled as a zero mean white noise process with covariance Q_m. If bounds on the magnitudes of maneuver are known, a method for choosing Q_m such that the actual performance of the filter is bounded by the computed filter covariance is described in [8].

Notice that Eq. (3.3) assumes x_m is temporally uncorrelated. This assumption may not always be very realistic. Methods for selecting a correlation model for aircraft tracking are given in [11]. Performance of techniques which employ the white noise acceleration model in re-entry vehicle tracking can be found in [7].

The maneuvering state x_m is usually not influenced by the state vector x. Under such a condition one can linearize the dynamic model and compute the state and maneuver estimates separately to obtain

$$\hat{x}_{k/k} = \tilde{x}_{k/k} + A_k \hat{x}_{m,k} \tag{3.4}$$

where $\tilde{x}$ is the estimate assuming x_m is zero, $\hat{x}_{m,k}$ is the maneuver estimate, A_k is a gain matrix, and $\hat{x}_{k/k}$ is the final state estimate. This decoupled implementation is computationally more efficient. It can be shown that the decoupled estimator is optimum for linear systems with known maneuvering time, see for example, [12].

Since the maneuver time is unknown, one has the problem of maneuver detection. If one applies the above method throughout the entire track interval the filter performance will be degraded when the target is non-maneuvering, [7], [8]. An ideal extension is to use a maneuver detector to switch between tracking filters.

3.3. Maneuver Detection

The detection problem involves discriminating between the following two hypotheses based upon filter residuals:

$$\begin{aligned} H_1: \ \tilde{\gamma}_k &= \gamma_k + g_k(x_m); \quad \text{maneuvering target hypothesis} \\ H_0: \ \tilde{\gamma}_k &= \gamma_k \qquad\qquad ; \quad \text{non-maneuvering hypothesis} \end{aligned} \tag{3.5}$$

where γ_k is the residual vector prior to initial maneuver, and g( ) is a known function which relates the maneuver vector, x_m, to the residual vector, $\tilde{\gamma}_k$.

The use of a generalized likelihood ratio test for maneuver detection is discussed in [7] for re-entry vehicles and in [13] for aircraft. A general discussion on the detection of sudden changes in linear systems can be found in [12].

Drawbacks to the use of detector-directed tracking include (1) detection delay and (2) large state vector estimate transient errors following the switch in filters. Numerical results in [7] show large estimation errors immediately after switching filters. This is due partially to the fact that the maneuver filter has yet to complete its start-up transient period.

Notice that maneuver detection relys on the fact that a target maneuver generates residual bias in a non-maneuvering filter. Upon switching to a filter which assumes the target can maneuver the above detection scheme cannot be used to discriminate if the target has returned to a non-maneuvering status.

All the above drawbacks can be alleviated if one employs an adaptive multiple model estimator.

3.4. Multiple Model Estimator

Let H_1 denote the hypothesis that the target is maneuvering and H_0 the hypothesis that the target is nonmaneuvering. One may construct two filters, the first uses a maneuvering dynamic model while the other assume a non-maneuvering dynamic model. Let these respective state estimates be denoted by $\hat{x}^1_{k/k}$ and $\hat{x}^0_{k/k}$. The optimum estimate $\hat{x}_{k/k}$ is obtained using

$$\hat{x}_{k/k} = P_1(k)\hat{x}_{k/k} + P_0(k)\hat{x}_{k/k} \tag{3.6}$$

where $P_i(k)$ is the a posteriori probability that H_i is true at time t_k.

The computation of $P_i(k)$ can be very complicated. If one uses a simple assumption that the hypotheses at time t_k are independent of hypotheses before t_{k-1} then a much simpler expression for $P_i(k)$ can be obtained. The details for the computation of $P_i(k)$ and $\hat{x}^i_{k/k}$ are presented in [14].

A higher level multiple model approach employs several hypotheses to model different maneuvering force levels. This approach enhances the estimation performance with the price of more computational resources.

We emphasize that the multiple model estimator is a general adaptive estimation technique which has found application in many areas and is under consideration for many others. A word of caution about the multiple model method; it is a highly tuned algorithm (for the case of modeling various maneuvering levels) and is therefore very vulnerable to situations in which none of the models match the actual dynamics. In such cases, erratic filter behavior may be observed. Such behavior is not yet completely understood from a theoretical point of view, [15].

4. TRACKING WITH ANGLE-ONLY MEASUREMENTS

Tracking with angle-only measurements constitutes a weakly observable system, i.e., the observability matrix (computed by using Jacobian matrices for nonlinear systems) is nearly singular. Such situations arise when either a radar or sonar is not feasible (limited by peak power, range, reverberations, or others) or the range estimate is not important (e.g., in the proportional navigation system). In this paper, we review the problem of ballistic target trajectory estimation with angle-only measurements.

4.1. Orbit Determination Problem

Typical situations in which one observes a ballistic target such as a satellite or a long range ballistic missile involve the use of a passive sensor which result in substantial measurement errors along the line-of-sight direction. The performance of the extended Kalman filter may become unacceptable as a result of such poor range information.

The orbital trajectory can usually be modeled by a set of deterministic differential equations of motion in which the process noise terms are zero. Under this condition, a least square (or maximum likelihood) estimator which iteratively solves for the optimum state estimate can be obtained, [16]. Furthermore, in such systems, only the initial state is required to define the entire trajectory. If $\hat{x}^k_1$ denotes the k-th iteration of the initial state estimate then the iterative algorithm given in [16] may be written

$$\hat{x}^{k+1}_1 = \hat{x}^k_1 + P^k_1\left[\sum_{n=1}^{N} \phi^{k^T}_n H^{k^T}_n \bar{R}^{-1}_n (z_n - h(\hat{x}^k_n))\right]$$

$$P^k_1 = \left[\sum_{n=1}^{N} \Phi^{k^T}_n H^{k^T}_n \bar{R}^{-1}_n H^k_n \Phi^k_n\right]^{-1} \tag{4.2}$$

where P^k_1 is the covariance of $\hat{x}^k_1$ and Φ^k_n is the linearized transition matrix evaluated at $\hat{x}^k_n$. Notice that the above algorithm is a batch processor. The $\hat{x}^k_n$ is obtained simply by numerical integration using $\hat{x}^k_1$ as the initial condition. More discussion of the above algorithm and related angle-only tracking issues can be found in [16].

5. TRACKING WITH MULTIPLE SENSORS

Some applications require multiple sensors operating in a multiple target environment. In this section we discuss the multiple sensor tracking problem and defer the multi-target problem to the next section.

Consider the situation in which there are several sensors simultaneously observing a single target. If these measurements are not time synchronized, one can formulate the problem as that of state estimation with non-uniform measurement times. It should be noted that in this particular formulation the functional transformation from state space to observation space depends upon the geometry, hence one can be confronted with a situation in which the measurement equations change from observation to observation.

If the measurements are time synchronized (such as in a multi-static radar system), then one is interested in seeking the most efficient way of processing the measurements. Three processing methods are discussed in [17].

(1) Parallel filter: The set of measurement vectors are concatenated to form a new measurement vector which is processed by the filter.

(2) Sequential filter: Each measurement is processed sequentially by the filter with zero prediction time between each measurement.

(3) Data compression filter: Prior to processing, measurement vectors are transformed to a common coordinate system and combined into a single measurement vector.

The above three methods can be shown to be equivalent and optimum for linear systems, [17]. The data compression method is computationally the most efficient for a wide range of cases [17].

It is possible to use decentralized processing wherein each sensor produces its own estimate. These are then combined to form a single estimate. The major drawback to such an approach is that the computational burden is significantly higher than any of the methods described above.

Algorithms for data compression with non-synchronized data vectors is a straightforward extension of the work described above. This may involve integration of measurement vectors to achieve time alignment followed by an application of the data compression equations. Polynomial data smoothing may be used for data compression in very high data rate systems as is discussed in [26]. One can also apply polynomial smoothers to the non-synchronized data case.

6. TRACKING IN A DENSE TARGET ENVIRONMENT

Tracking in a dense target environment was a subject of a recent review article [18] and the theme of a session in the recent IEEE-CDC conference [19]. Representative literature on this topic can be found in [18]-[25]. This problem is often referred to as the scan-to-scan correlation or tracking data association.

This problem can be separated into two stages. The first stage involves track initiation and the second stage involves track maintenance. They are discussed individually below.

6.1. Track Initiation

Consider the case of a scanning sensor. The first and second scan produce N_1 and N_2 detections, respectively. The question is, how does one associate the two sets of detections N_1 and N_2 to form a max (N_1, N_2) number of track files? Notice that we have assumed that $N_1 \neq N_2$ which can be the result of either (1) an imperfect detection process or (2) the emergence of new targets during the time between the two scans. In the following, we describe an approach for track initiation which uses k scans of data.

Let Z_k denote the total measurements (N) collected at kth scan, i.e.,

$$Z_k = \{z_1(k), z_2(k), \ldots, z_N(k)\} \tag{6.1}$$

We use Z_k to denote the set of measurements up to and including kth scan, i.e.,

$$Z^k = \{Z_i;\ i=1, \ldots, k\} \tag{6.2}$$

Furthermore, to simplify discussion let us assume that the number of detections is N for all Z_k's. When this is not true, one must enumerate additional hypotheses to account for all possibilities. With Z^k, there are N^k potential combination of measurement sequences which would represent a possible track. Let each combination be denoted by the hypothesis $H_{m_k}(k)$ which is defined by

$$H_{m_k}(k) = \{z_{n_1}(1), z_{n_2}(2), \ldots, z_{n_k}(k)\} \tag{6.3}$$

Applying Kalman filters to process each possible measurement sequence one can obtain the a posteriori probability of the hypothesis H_{m_k} being true. This measure can be computed recursively using

$$P(H_{m_k}(k)/Z^k) = \frac{p(z_{n_k}(k)/H_{m_{k-1}}(k-1), Z^{k-1})}{p(z_{n_k}(k)/Z^{k-1})} P(H_{m_{k-1}}(k-1)/Z^{k-1}) \tag{6.4}$$

where $p(z_{n_k}(k)/H_{m_{k-1}}(k-1), Z^{k-1})$ is the density of the innovation sequence from the Kalman filter using $H_{m_{k-1}}(k-1)$ and $z_{n_k}(k)$. The above equation can be derived as a special case of the results in [14]. The final set of tracks (total N) are chosen as those N feasible hypotheses having the largest a posteriori probabilities, i.e.,

$$\max_{\{N:\ H_{m_k}(k) \in \mathscr{F}\}} \{P(H_{m_k}(k)/Z^k);\ m_k = 1, \ldots, N^k\} \tag{6.5}$$

where the feasible set $\mathscr{F}$ restricts candidate solutions such that each measurement at a given time is used only once.

The computational requirement of the above method is clearly non-trivial. In many applications, it is possible to pre-cluster the detections so that search over the entire set of detections is unnecessary. Other physical reasons peculiar to given systems or applications can sometimes be exploited to reduce the search requirements.

A similar approach using a maximum likelihood method is described in [21]. Other track initiation techniques are contained in [20], [25].

6.2. Track Maintenance

Once track files have been established, the computational needs are greatly reduced. This occurs since each track file need only search a much smaller "admissible" region of space determined by the covariance of the filter residual process.

Note that a slightly modified method of the track initiation algorithm discussed in 6.1 can be applied to each track file. In this case one would propagate a new track for each detection resident in the admissable region of the observation space. This procedure results in a number of track files which grows exponentially. One can inhibit the growing memory and computational requirements to a maximum by arbitrarily selecting a tree depth and conducting a global search for a set of feasible tracks having the highest hypothesis probabilities. Another approach is to combine a set of "most likely hypotheses" growing out of the same track file using the weighted sum of state estimates with the hypothesis probabilities as weighting factors. This second approach is the basis of the Bayesian tracker presented by Singer et al [22], [23]. If the depth is equal to unity, i.e., one combines all admissible detections at each scan, then the probabilistic data association techniques of Bar-Shalom and Tse [24] are obtained.

We have not explicitly discussed the problems of track termination, track with new initiations, false measurement, etc. An interesting paper which combines both features of track initiation and track maintenance while providing an extensive discussion on many related problems is Ref. [25]. We note that research on this subject area is currently underway at a number of organizations and many problems have yet to be addressed.

7. CONCLUDING REMARKS

In this paper, we have presented a survey of problems and solutions for target tracking. Although this problem has been actively dealt with by practitioners for both military and civilian applications over many years, new problems still emerge and challenge system engineers. It is felt that while the basic tracking filter design is no longer an issue, the problems of tracking in more stressing situations and the integration of tracking algorithms with the overall command, control, and communication systems are still of current interest.

REFERENCES

[1] R. P. Wishner, R. E. Larson, and M. Athans, "Status of Radar Tracking Algorithms" Proceedings of Symposium on Nonlinear Estimation Theory and Its Applications, San Diego, California, Sept. 1970, pp 32-54.

[2] M. Gruber, "An Approach to Target Tracking" M.I.T., Lincoln Laboratory, Technical Note 1967-8, (10 February 1971) DDC-AD-654272.

[3] R. K. Mehra, "A Comparison of Several Nonlinear Filters for Re-entry Vehicle Tracking," IEEE Trans. on Automatic Control, Vol. AC-16, August 1971, pp 307-319.

[4] R. A. Singer and K. W. Behnke, "Real-time Tracking Filter Evaluation and Selection for Tactical Application," IEEE Trans. on Aerospace and Electronic Systems, Vol. AES-7, January 1971, pp 100-110.

[5] T. R. Benedict and G. W. Bordner, "Synthesis of an Optimal Set of Radar Track-while-scan-Smoothing Equations, IRE Trans. on Automatic Control, Vol. AC-7, July 1962, pp 27-32.

[6] S. R. Neal, "Discussion on Parametric Relations for α-β-γ Filter Prediction, IEEE Trans. on Automatic Control, Vol. AC-12, June 1962, pp 315-317.

[7] C. B. Chang, R. H. Whiting and M. Athans, "On the State and Parameter Estimation for Maneuvering Re-entry Vehicles," IEEE Trans. on Automatic Control, Vol. AC-22, February 1977, pp 99-105.

[8] C. B. Chang and K. P. Dunn, "Kalman Filter Compensation For A special Class of Systems," IEEE Trans. on Aerospace and Electronic System, Vol. AES-13, Nov. 1977, pp 700-706

[9] T. J. Tarn and J. Zaborszky, "A Practical, Nondiverging Filter," AIAA Journal, Vol. 8, June 1970, pp 1127-1133.

[10] A. H. Jazwinski, "Adaptive Filtering," Automatic, Vol. 5, 1969, pp 475-485.

[11] R. A. Singer, "Estimating Optimal Tracking Filter Performance for Manned-Maneuvering Targets," IEEE Trans. on Aero. and Elec. Syst., Vol. AES-6, July 1970, pp 473-483.

[12] C. B. Chang and K. P. Dunn, "On GLR Detection and Estimation of Unexpected Inputs in Linear Discrete Systems," IEEE Trans. on Auto. Control, Vol. AC-24, June 1979, pp 499-501.

[13] R. J. McAulay and E. Denlinger, "A Decision-Direction Adaptive Tracker," IEEE Trans. on Aero. and Elec. Syst., Vol. AES-9, March 1973, pp 229-236.

[14] C. B. Chang and M. Athans, "State Estimation for Discrete Systems with Switching Parameters," IEEE Trans. on Aero. and Elec. Syst., Vol. AES-14, May 1978, pp 418-425.

[15] M. Athans, et al., "The Stochastic Control of the F-8C Aircraft Using a Multiple Model Adaptive Control (MMAC) Method - Part I: Equilibrium Flight," IEEE Trans. on Automatic Control, Vol. AC-22, October 1977, pp. 768-779.

[16] C. B. Chang, "Ballistic Trajectory Estimation with Angle-Only Measurements, The 18th IEEE Conference on Decision and Control, Fort Lauderdale, Fla., December 1979.

[17] D. Willner, C. B. Chang, and K. P. Dunn, "Kalman Filter Algorithms for a Multi-sensor System," 1976 IEEE Conf. on Decision and Control, Clearwater, Fla. December 1976.

[18] Y. Bar-Shalom, "Tracking Methods in a Multitarget Environment" IEEE Trans. on Automatic Control, Vol. 23, August 1978, pp 618-626.

[19] Proceedings of the 1978 IEEE Conf. on Decision and Control, San Diego, CA, pp 1182-1216.

[20] R. W. Settler, "An Optimal Data Association Problem in Surveillance Theory," IEEE Trans. on Military Electronics Vol. MIL-8, April 1964, pp 125-139.

[21] C. L. Morefield, "Application of 0-1 Integer Programming to Multitarget Tracking Problems," IEEE Trans. on Automatic Control, Vol. AC-18, Dec. 1973, pp 571-581.

[22] R. A. Singer and R. G. Sea, "New Results in Optimizing Surveillance System Tracking and Data Correlation Performance in Dense Multitarget Environments," IEEE Trans. on Automatic Control, Vol. AC-18, Dec. 1973, pp 571-581.

[23] R. A. Singer, R. G. Sea, and K. Housewright, Derivation and Evaluations of Improved Tracking Filters for Use in Dense Multitarget Environment," IEEE Trans. on Inf. Theory, Vol. IT-20, July 1974, pp 423-432.

[24] Y. Bar-Shalom and E. Tse, "Tracking in a Cluttered Environment with Probabilistic Data Association," Automatica, Vol.-11, Sept. 1975, pp 451-460

[25].D. B. Reid, "An Algorithm For Tracking Multiple Targets," 1978 IEEE Conf. on Decision and Control, San Diego, CA, pp 1202-1211.

[26] Y. Bar-Shalom, "Redundancy and Data Compression in Recursive Estimation," IEEE Trans. on Automatic Control, Vol. 17, October 1972, pp 684-689.

ROBUST FILTERING AND SMOOTHING OF TRACKING DATA

WILLIAM S. AGEE and ROBERT H. TURNER
Mathematical Services Branch
Analysis and Computation Division
White Sands Missile Range, NM 88002

ABSTRACT

Robust methods provide a fresh approach to the problem of treatment of wild observations in filtering and smoothing problems. The robust M-estimates of regression are extended to filtering and fixed lag smoothing employing a pseudo-density of the observations in a conditional mean derivation of the filter and fixed lag smoother. These robust methods have been applied to simulated and real tracking data to obtain improved estimation performance in the presence of wild observations.

INTRODUCTION

Robust filtering and smoothing are a natural extension of the robust M-estimates of regression developed by Huber [1]. The robust M-estimates provide a natural treatment of outlying observations and have been extremely successful in dealing with outliers in other data reduction problems [2] and [3]. The extension of the M-estimate methods provides a fresh approach to the problems caused by outliers in filtering and smoothing applications. Robust methods for estimation are designed to perform well when observations from contaminating distributions are present. The conventional estimation techniques of least squares, maximum likelihood, minimum variance, etc. may become useless when the observations are contaminated by gross outliers or wild data points. When using these estimation methods, outliers are often treated by testing the residuals. If it is decided that a residual is statistically too large, the corresponding observation is declared an outlier and is not processed. These hypothesis testing methods are often successful if only a small number of outliers are present but breakdown for larger proportions of outlying observations. Also, in order for outliers to be detected, they must be relatively large compared to the measurement noise. The detection methods based on testing of residuals are relatively insensitive to small outliers which leads to an inflation of the mean square estimation error. Thus, methods for treating outliers should be evaluated on their ability to achieve a small mean square estimation error as well as their ability to offer protection from gross outliers.

Very little development has appeared on the application of robust estimation techniques to filtering and smoothing. The most notable work in this direction is that of Masreliez and Martin [4]. Their development of the application of M-estimates to the Kalman filter is mainly theoretical. The emphasis here is on the development of some practical results on the application of M-estimates to filtering and smoothing. We have applied these methods to filtering and smoothing of tracking data from trajectory measurement systems at WSMR. Using simulated tracking data we have also performed extensive Monte Carlo evaluation of filtering methods based on M-estimates to determine the conditions for which we can expect to benefit from the application of these methods.

M-ESTIMATES FOR REGRESSION

Given scaler observations y_i, i = 1, N of a linear model

$$y_i = X_i\Theta + e_i \tag{1}$$

where X_i is a row vector of known independent variables and e_i is a random error term we want to estimate the unknown p-vector, Θ. The M-estimate of Θ minimizes

$$\sum_{i=1}^{N} \rho\left((y_i - X_i\Theta)/s\right), \tag{2}$$

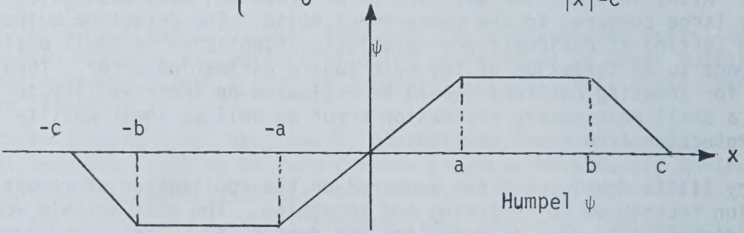
where ρ(·) is a specified function and s is a robust measure of dispersion of the residuals, $y_i - X_i\Theta$. Minimizing (2) by differentiating with respect to Θ gives

$$\sum_{i=1}^{N} X_i^T \psi\left((y_i - X_i\hat{\Theta})/s\right) = 0 \tag{3}$$

where ψ is the derivative of ρ and $\hat{\Theta}$ is the M-estimate of Θ. (3) is the analog of the normal equations in least squares estimation. Θ is computed iteratively by applying a weighted least squares algorithm to (3). For details see [2].

Rather than specifying the function ρ, M-estimates are usually described by specifying the function ψ. Several ψ functions have been proposed in the literature. The only ψ functions considered here are variations of the one proposed by Hampel [5]. The Hampel ψ function with breakpoints a, b, c, denoted by Ha(a,b,c) is given by

$$\psi(x) = \begin{cases} x & |x| \le a \\ a\cdot\mathrm{sgn}(x) & a \le |x| \le b \\ a(x - c\cdot\mathrm{sgn}(x))/(b - c) & b \le |x| \le c \\ 0 & |x| \ge c \end{cases} \tag{4}$$



Humpel ψ

The M-estimates can also be applied to regression problems having vector observations and to nonlinear regression problems. If the probability density function p of the observations is related to ψ by $p'/p = -\psi$, the resulting M-estimate is maximum likelihood. For any ψ function we call $\bar{e}^{\rho}$ a pseudo-density and derive filters in some conventional ways with the density function of the observations replaced with a pseudo-density.

APPROPRIATE NON-GAUSSIAN FILTERING

Assume that the state x(k) of the process being observed is governed by the discrete linear model,

$$x(k+1) = \Phi(k+1,k)\, x(k) + u(k) \tag{5}$$

where the state vector $x(k)$ is an n-vector, $u(k)$ is a Gaussian state noise vector with zero mean and covariance $Q(k)$. $\Phi(k+1,k)$ is an $n \times n$ state transition matrix. Scalar observations, $z(k)$, of the process are given by

$$z(k) = H(k)x(k) + v(k) \tag{6}$$

where $H(k)$ is a row vector and $v(k)$ is a measurement noise error which may be contaminated by outliers.

In order to derive robust filters corresponding to the M-estimates of regression, we use the results of Masreliez [6] on approximate non-Gaussian filtering. Masreliez obtains an approximate conditional mean of $p(x(k)|Z^k)$ where Z^k is the collection of observations, $Z^k = \{z(1), z(2), ---, z(k)\}$. Using Bayes rule $p(x(k)|Z^k)$ is given by

$$p(x(k)|Z^k) = \frac{p(z(k)|x(k))p(x(k)|Z^{k-1})}{p(z(k)|Z^{k-1})} \tag{7}$$

In order to derive a useful approximation to the conditional mean, $E[x(k)|Z^k]$, of (7), Masreliez assumes that $p(x(k)|Z^{k-1})$ is Gaussian with mean $\hat{x}(k|k-1)$ and covariance $P(k|k-1)$. The resulting approximate conditional mean, $\hat{x}(k|k)$ is given by

$$\hat{x}(k|k) = \hat{x}(k|k-1) + P(k|k-1)H^T(k)g(z(k)), \tag{8}$$

where $g(z(k)$ is the scalar

$$g(z(k)) = -p^{-1}(z(k)|Z^{k-1})\frac{\partial}{\partial z(k)}p(z(k)|Z^{k-1}) \tag{9}$$

Masreliez also derives the second moment, $P(k|k)$, of $p(x(k)|Z^k)$. The result is

$$P(k|k) = P(k|k-1) - P(k|k-1)H^T(k)G(z(k))H(k)P(k|k-1), \tag{10}$$

with

$$G(z(k)) = \frac{\partial g(z(k))}{\partial z(k)} \tag{11}$$

A second method for approximate non-Gaussian filtering is the marginal maximum likelihood filter. In this case we find the estimate $\hat{x}(k|k)$ which maximizes (7). In this derivation we also assume that $p(x(k)|Z^{k-1})$ is Gaussian. The resulting equation for $\hat{x}(k|k)$ is

$$\hat{x}(k|k) = \hat{x}(k|k-1) - P(x|k-1)H^T(k)p^{-1}(z(k)|\hat{x}(k|k))\frac{\partial p(z(k)|\hat{x}(k|k))}{\partial z(k)} \tag{12}$$

The estimate given by (12) is of the same form as the conditional mean estimate given by (8). The difference in the two estimates is that the right hand side of (12) depends on $\hat{x}(k|k)$ while the right hand side of (8) depends only on the predicted estimate, $x(k|k-1)$. Thus, (12) requires iteration to obtain the estimate and if in the first iteration of (12), we substitute $\hat{x}(k|k-1)$ for $\hat{x}(k|k)$ on the right hand side, the resulting estimate on the first iteration will be identical to the conditional mean estimate of (8). Thus, the maximum likelihood filter may be regarded as a correction to the conditional mean filter.

ROBUST FILTER EQUATIONS

The robust filtering equations corresponding to M-estimates are obtained by replacing the density function, $p(z(k)|Z^{k-1})$, in (9) or (12) by a pseudo-density, $e^{-\rho}$, where $\rho' = \psi\left(\frac{z(k) - H(k)\hat{x}(k|k-1)}{s}\right)$ is a desired influence function. This substitution results in the following equations for the conditional mean robust filter.

$$\hat{x}(k|k) = \hat{x}(k|k-1) + \frac{P(k|k-1)H^T(k)}{s_k}\,\psi\left(\frac{z(k) - H(k)\hat{x}(k|k-1)}{s_k}\right) \tag{13}$$

$$P(k|k) = P(k|k-1) - \left(\psi'\left(\frac{r(k)}{s_k}\right)\Big/ s_k^{\,2}\right)P(k|k-1)H^T(k)H(k)P(k|k-1), \tag{14}$$

where ψ' is the derivative of ψ and $r(k) = z(k) - H(k)\hat{x}(k|k-1)$. The filter equations are completed by the usual Kalman filter equations for the predicted moments.

In order to insure the robustness of the filter described by (13) and (14), the dispersion s_k of the predicted residuals must be specified so that it is insensitive to outliers. We used the MAD estimate of s_k computed from past residuals as

$$s_k = \underset{j=0,N-1}{\text{median}} \left| z(k-j) - H^T(k-j)\hat{x}(k-j|k-j-1) \right| \Big/ .6745 \tag{15}$$

where N is a suitably chosen integer.

The robust maximum likelihood filter is obtained by replacing the density $p(z(k)|Z^{k-1})$ in (12) by a pseudo-density. The resulting filter is given by

$$\hat{x}(k|k) = \bar{x}(k|k-1) + \frac{P(k|k-1)H^T(k)}{s_k}\;\psi\left(\frac{z(k) - H(k)\hat{x}(k|k)}{s_k}\right) \tag{16}$$

In (16) we use $\bar{x}(k|k-1)$ to denote the mean of $p(x(k)|Z^{k-1})$. We use (14) to compute $P(k|k)$ and use (13) to compute $\bar{x}(k|k)$. Several simple methods are available for the iterative solution of (16). The simplest of these is to use

$$\hat{x}^{(\alpha+1)}(k|k) = \bar{x}(k|k-1) + \frac{P(k|k-1)H^T(k)}{s_k}\;\psi\left(\frac{z(k) - H(k)\hat{x}^{(\alpha)}(k|k)}{s_k}\right), \tag{17}$$

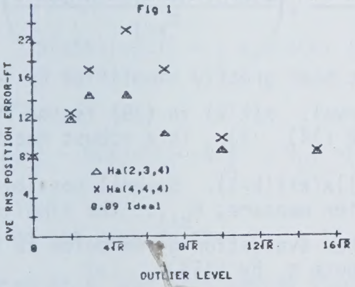
starting with $\hat{x}^{(0)}(k|k) = \bar{x}(k|k-1)$ so that the first estimate $\hat{x}^{(1)}(k|k)$ is the conditional mean estimate, $\bar{x}(k|k)$.

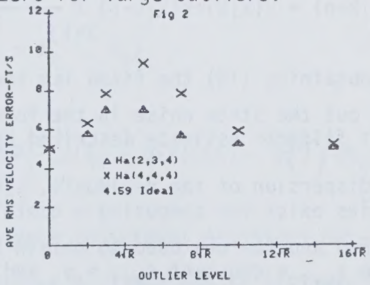
EVALUATION OF THE ROBUST FILTER

Evaluation of the robust filtering methods described above was done with a view toward eventual application to trajectory estimation. The emphasis in the evaluation was on simulated rather than real trajectory data. This allows a quantitative determination of any advantages in the use of robust filtering in the presence of outliers and also any loss in efficiency using robust methods when no outliers are present. The simulated trajectory is that of a constant velocity, level flying aircraft. The measurements are of position in each of three cartesian components with added measurement noise which is contaminated by outliers. The filter model assumes the trajectory to have constant acceleration in each coordinate. The filter for each coordinate has a small acceleration state noise, Q(k) = 5. The outlier contamination is controlled by a two state Markov chain with a transition possible at each measurement time. An outlier is added to the measurement if the Markov chain is in state two and no outlier is added if the chain is in state one. The transition probabilities, Pij, are used to determine the percentage of outliers contaminating the measurements and also the length of runs of outliers in the measurements. The magnitude of the outlier contamination is $C \cdot \sqrt{R}$, i.e., a constant multiple of the measurement noise standard deviation.

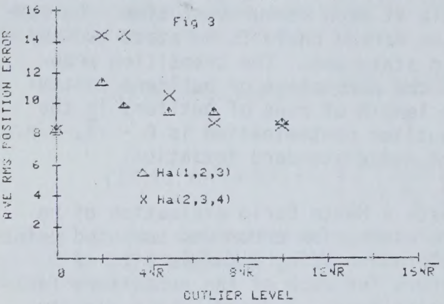
Using the simulated trajectory data a Monte Carlo evaluation of robust filtering was performed. The rms estimation error was computed pointwise for position, velocity, and acceleration using a sample size of twenty-five. The plots of the rms errors for each of the conditions tested requires far too much space to present here. Instead, these results are summarized by time averages of the rms error in position and velocity for each of the conditions tested.

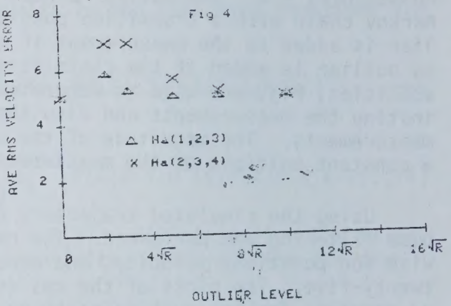
Figure 1 compares the average rms position error for two filters using the Hampel ψ functions Ha(2, 3, 4) and Ha (4, 4, 4). Figure 2 gives the rms velocity error comparison for the same two filters. Also indicated in Figures 1 and 2 are the ideal rms error values which were obtained with an ordinary Kalman filter with no outliers present and using a known measurement covariance, $R_k = 400$. The Monte Carlo evaluation of figures 1 and 2 was made with a measurement noise standard deviation of $\sqrt{R_k} = 20$ ft. R_k was unknown to the filter.
We note from figures 1 and 2 that neither of the robust filters lose much efficiency from the ideal values when no outliers are present. The error curves in figures 1 and 2 behave as expected. Since outliers small in relation to the measurement noise are hardest to detect, the error curve rises sharply. Outliers large relative to the measurement noise are easy to detect so the error curve returns to zero for large outliers.





We see from figures 1 and 2 that Ha (2, 3, 4) has a significantly smaller mean square error than Ha (4, 4, 4). Except for the way in which the dispersion of the residuals is measured, i.e., the MAD estimate in (15), Ha (4, 4, 4) is a conventional way of handling outliers in a Kalman filtering application. Using Ha (4, 4, 4) any observation whose predicted residual is greater than 4 • s is not processed and any observations whose predicted residual is less than 4 • s is processed as an ordinary Kalman filter observation. The above Monte Carlo evaluation was made with Markov chain probabilities $P_{21} = .05$ and $P_{12} = .5$ which gives an outlier probability of.088 and an average outlier run length of three. In order to reduce the average rms errors, we pull in the breakpoints of the Hampel ψ function. Figures 3 and 4 compare the average rms errors in position and velocity **for** the Ha (1, 2, 3) and Ha (2, 3, 4).





Figures 3 and 4 were made using the same outlier proportions and measurement noise standard deviations as figures 1 and 2.

The iterated filter, i.e., the approximate maximum likelihood filter given by (14), (16), and (17) was also evaluated under the same conditions as the conditional mean filter. Comparison of the average mean square errors for the conditional mean and maximum likelihood filters showed no discernible differences.

ROBUST SMOOTHING

A simplified robust, fixed lag smoother, was derived in a similar manner to the robust conditional mean filter derivation. We derive an approximate conditional mean, $E[x(k)|Z^{k+n}]$, of the density $p(x(k)|Z^{k+n})$ using the same methods and assumptions used by Masreliez and used in the derivation of the robust filter. The robust smoothed estimate is given by

$$\hat{x}(k|k+n) = \hat{x}(k|k)+P(k|k-1) \sum_{j=1}^{n} \frac{\Phi^T\left(t_{k+j},t_k\right)H^T(k+j)}{s_{k+j}} \cdot \psi\left(\frac{Z(k+j)-H(k+j)\hat{x}(k+j|k-1)}{s_{k+j}}\right) \quad (18)$$

In obtaining (18) the fixed lag smoother has been greatly simplified by leaving out the state noise in the forward interval. $x(k|k)$ in (18) is the robust filtered estimate described by (13) and (14). s_{k+j} is a robust measure of dispersion of the residuals, $z(k+j)-H(k+j)\hat{x}(k+j|k-1)$. Several possibilities exist for computing a useful dispersion measure, s_{k+j}. The simplist method and the one used to obtain the smoother evaluation given below is to make s_{k+j} a constant $s_{k+j} = s_k$ and then compute s_k by (15).

A Monte Carlo evaluation of the simplified fixed lag smoother was performed using the same simulated trajectory as was used for the filter evaluation. Measurement noise having a standard deviation of 50 feet was added to the simulated positions. The measurement noise standard deviation was unknown to the smoother. The forward smoothing interval had a length of n = 20 which represents a one second smoothing time. The outlier proportions and run lengths were the same as for the filter evaluation. A sample size of ten was used for the Monte Carlo evaluation of the smoother.

Figures 5 and 6 display the average rms position and velocity estimation errors obtained using the robust, fixed lag smoother with the Hampel ψ functions, Ha (2, 3, 4) and Ha (4, 4, 4). Also noted in figures 5 and 6 is the ideal average rms values which were obtained using the smoother with no outliers and a known measurement covariance, $R_k = 2500$. The robust smoother using Ha (4, 4, 4) is representative, except for the measurement of s_k, of a conventional way of handling outliers in an optimal smoothing application. We note that either of the smoothers offers good protection from very large outliers but that Ha (2, 3, 4) results in a significantly smaller estimation error when small outliers are present.

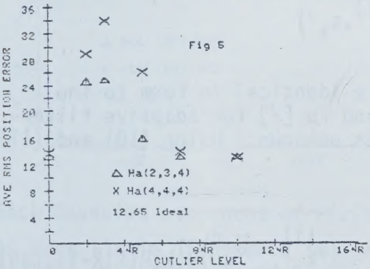


Fig 5

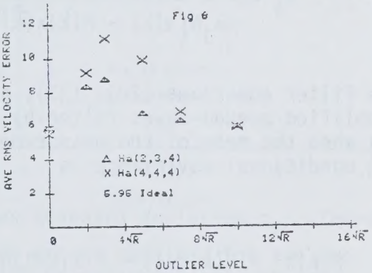


Fig 6

ROBUST FILTERING USING GAUSSIAN MIXTURES

Robust estimation methods deweight and/or reject outlying observations by specifying observational densities or pseudo-densities having long, heavy tails. We can also realize a long, heavy tailed density by using a Gaussian mixture. Suppose we replace $p(z(k)|x(k))$ in (7) or (12) by the Gaussian mixture pseudo-density,

$$p(z(k)|x(k)) = \sum_i \alpha_i N(z(k) - H(k)x(k) - a_k^{(i)}, R_k) \tag{19}$$

where

$$N(z(k) - H(k)x(k) - a_k^{(i)}, R_k) = \left(1\Big/\sqrt{2\pi R_k}\right) \mathrm{EXP}\left\{-\left(z(k) - H(k)x(k) - a_k^{(i)}\right)^2 \Big/ 2R_k\right\} \tag{20}$$

We do not require that $\sum_i \alpha_i = 1$. Thus, we have individual Gaussians centered at $a_k^{(i)}$ and each having standard deviation R_k. The sum in (19) may be infinite. The locations, $a_k^{(i)}$, and the amplitudes, α_i, are free parameters. Using (19) we obtain $p(z(k)|Z^{k-1})$ as

$$p(z(k)|Z^{k-1}) = \sum_i \alpha_i N\left(z(k) - H(k)\hat{x}(k|k-1) - a_k^{(i)}, H(k)P(k|k-1)H^T(k) + R_k\right) \qquad (21)$$

In obtaining (21) we have again assumed that $p(x(k)|Z^{k-1})$ is Gaussian.

Substituting (21) into (8) and (9) with the variance of the residuals, $H(k)P(k\ k-1)H^T(k)+R_k$, replaced with an estimated value, s_k^2, gives

$$\hat{x}(k|k) = \hat{x}(k|k-1) + \frac{P(k|k-1)H^T(k)}{s_k^2}\left(z(k) - H(k)\hat{x}(k|k-1) - \bar{a}_k\right) \qquad (22)$$

In (22) $\bar{a}_k$ is the weighted average

$$\bar{a}_k = \sum_i W_i a_k^{(i)}, \qquad (23)$$

where the weights W_i are given by

$$W_i = \frac{\alpha_i N\left(z(k) - H(k)\hat{x}(k|k-1) - a_k^{(i)}, s_k^2\right)}{\sum_j \alpha_j N\left(z(k) - H(k)\hat{x}(k|k-1) - a_k^{(j)}, s_k^2\right)} \qquad (24)$$

The filter equations (22), (23), and (24) are identical in form to the simplified pseudo-Bayes filter by Ackerson and Fu [7] for adaptive filtering when the mean of the measurement noise is unknown. Using (10) and (11) the conditional covariance is

$$P(k|k) = P(k|k-1) - P(k|k-1)H(k)\left({}^{1}\!/_{s_k^2} - \frac{\overline{\left(a_k^{(i)} - \bar{a}_k\right)^2}}{s_k^4}\right)H^T(k)P(k|k-1), \qquad (25)$$

where

$$\overline{\left(a_k^{(i)} - \bar{a}_k\right)^2} = \sum_i W_i \left(a_k^{(i)} - \bar{a}_k\right)^2 \qquad (26)$$

Although the above sums may be theoretically infinite, we only need to calculate the relatively few terms in the sums which have significant values of the weights, W_i. Thus, we compute only those terms in the sums for which $|z(k)-H(k)\hat{x}(k|k-1)-a_k^{(i)}|/s_k \leq 4$. With this simplification the amount of extra computation required to implement this robust filter is relatively small. The locations, $a_k^{(i)}$, produce a smooth pseudo-density if they are chosen as zero and odd integral multiples of s_k, $a_k^{(0)} = 0$, $a_k^{(i)} = \mathrm{sgn}(i)(2|i| - 1)s_k$, $|i| \geq 1$. We have also tested the filter with $a_k^{(i)} = i \cdot k$, $|i| \geq 0$. The value of the residual dispersion, s_k, is still computed by (15). Several different choices of the amplitudes have been tested. The most extensive testing has been done with $\alpha_i = 1$ and $\alpha_i = {}^{1}\!/(|i| + 1)$.

Some robust filters using the Gaussian mixture formulation were also evaluated via Monte Carlo testing. These filters were tested using the same simulated trajectory data and under the same outlier and measurement noise conditions as the robust filters using Hampel ψ functions. The sample size for Monte Carlo was twenty-five, the Markov transition probabilities were $P_{21} = .05$ and $P_{12} = .5$ and the measurement noise standard deviation, which was unknown to the filter and estimated by (15), was $\sqrt{R_k} = 20$ feet. Figures 7 and 8 present the average rms position and velocity errors for a Gaussian mixture filter with observations contaminated by various magnitudes of outliers. The Gaussian mixture filters used in generating Figures 7 and 8 used magnitudes of the Gaussians, $\alpha_i = 1/(|i| + 1)$. Two different Gaussian mixture filters are represented in Figures 7 and 8, one with Gaussians at all integral multiples of s_k and one with Gaussians at zero and odd integral multiples of s_k.

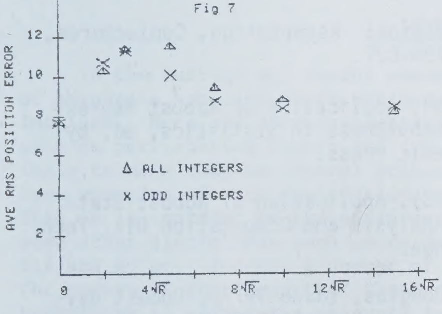


Fig 7

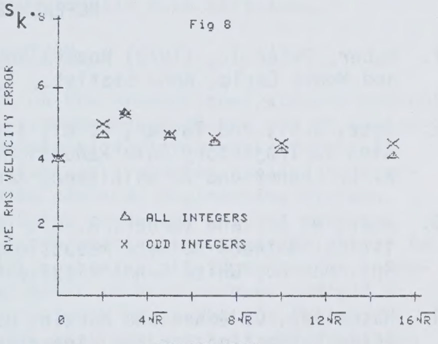


Fig 8

Each Gaussian component of $P(z(k)|Z^{k-1})$ has standard deviation s_k. There is very little difference in the estimation errors obtained for the two filters of Figures 7 and 8. The filter with Gaussians at only the odd multiples of s_k is computationally less complex. Figures 9 and 10 give the results of the Monte Carlo evaluation of a Gaussian mixture filter which places Gaussians at zero and odd integral multiples of s_k with amplitudes, $\alpha_i = 1$.

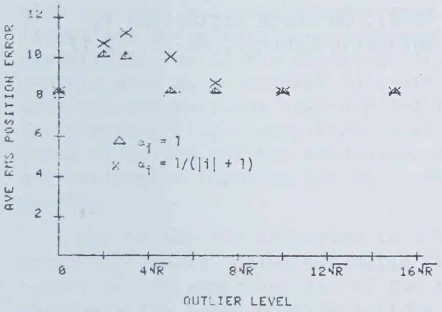


Fig 9

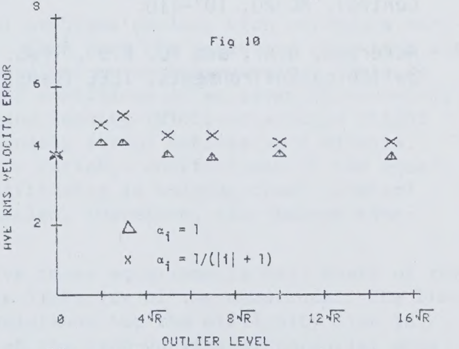


Fig 10

This filter appears to give somewhat smaller errors than the other Gaussian mixture filters and also slightly smaller errors that the robust filter which uses Ha(1, 2, 3). We note that using $\alpha_i = 1$ with Gaussians at all integral multiples of s_k does not result in a useful filter since it has a zero influence function and therefore does not produce any error correction.

CONCLUSIONS

Two methods based on M-estimates have been presented for robust filtering and smoothing, one using the Hampel ψ function with various breakpoints and the other which models the observation error as a Gaussian mixture. These robust filtering methods were subjected to a Monte Carlo evaluation using simulated trajectory data from an aircraft tracking application. The results of this evaluation show that both of these robust filtering methods give a significant reduction in average rms estimation error for small outliers compared to a more conventional way of treating outliers in an optimal filtering application. The tests also suggest that the Gaussian mixture robust filter methods offer the most promise for application and future testing. Further evaluation of robust filtering methods under more severe trajectory applications are necessary.

REFERENCES

1. Huber, Peter J., (1973) Robust Regression: Asymptotics, Conjectures, and Monte Carlo, Ann. Statist., 1, 799-821.

2. Agee, W.S., and Turner, R. H., (1979), Application of Robust Regression to Trajectory Data Reduction, Robustness in Statistics, ed. by R. L. Launer and G. Wilkinson, Academic Press.

3. Agee, W. S., and Turner, R. H., (1978), Application of Robust Statistical Methods to Data Reduction, Analysis and Computation Div. Tech. Rpt. No. 65, White Sands Missile Range.

4. Masreliez, C. Johan and Martin, R. Douglas, (June 1977), Robust Bayesian Estimation for the Linear Model and Robustifying the Kalman Filter, IEEE Trans. Automatic Control, AC-22, 361-371.

5. Hampel, Frank R., (June 1974). The Influence Curve and its Role in Robust Estimation, J. Amer. Statist. Assoc., 69, 383-393.

6. Masreliez, C. Johan, (Feb. 1975), Approximate Non-Gaussian Filtering with Linear State and Observation Relations, IEEE Trans. Automatic Control, AC-20, 107-110.

7. Ackerson, G.A., and Fu, K.S., (Feb. 1970), On State Estimation in Switching Environments, IEEE Trans. Automatic Control, AC-15, 10-17.

APPROXIMATE ANALYSIS OF VARIABLE LINEAR FILTERS FOR AEROSPACE APPLICATIONS

RUDRAPATNA V. RAMNATH and W. MICHAEL BOWLES
The Charles Stark Draper Laboratory, Inc.
Cambridge, Massachusetts 02139

ABSTRACT

Approximations to the solutions of a class of linear time varying systems are discussed. The Multiple Scales approach is briefly discussed and approximations by this method are presented both for scalar and vector-matrix forms of the mathematical model of the system. A variable estimation problem is presented with a comparison of the multiple scales and numerical solutions from the point of accuracy and computation time. In this study the approximations are computationally more efficient.

1. INTRODUCTION

In the current and recent research on the dynamic analysis and control of aerospace vehicles there has been a growing trend in the use of approximations. More and more systematic methods of deriving approximations, such as perturbation theory and its variations are being applied increasingly to many complex control problems in advanced engineering systems. Some examples of such applications are being presented in this meeting. This is in contrast to the early use of systematic approximation theory in some other disciplines such as celestial mechanics, fluid mechanics, physics and so on. Indeed, a number of important techniques have emerged in the course of development of these disciplines. Perturbation theory and techniques of asymptotic analysis form an important part of approximation theory. Applications to control theory are relatively recent and have largely consisted of some particular kinds of techniques (e.g. boundary layer methods) applied to some classes of problems (e.g. with fast and slow variables by Kokotovic and others [1]. In this paper we will discuss some rather simple approximations in the context of variable linear filters.

2. TIME VARYING SYSTEMS

A number of dynamics and control problems dealing with aerospace vehicles are nonlinear and time varying. Even when the equations of motion are linearized the resulting equations turn out to be time-varying. This is essentially due to the changing flight conditions of an advanced aerospace vehicle such as a vertical take-off and landing (VTOL) or a large flight envelope conventional take-off and landing (CTOL) vehicle or a missile. The changing flight conditions lead to variable coefficients in the equations of motion and the consequent difficulty in solving them. Control and estimation problems for such vehicles, therefore, also become time-varying.

Anyone who has attempted to solve these equations is well aware of the great difficulty, notwithstanding the linearity of the equations. The linearity allows superposition of the solutions but the difficulty lies in obtaining the basic eigen-solutions of the time-varying differential equations. Indeed, it is known that it is impossible to solve exactly the linear variable differential equations of second (and higher) order, in general. Therefore, in order to glean some degree of understanding and insight into the behavior of such systems, one has to resort to approximations. They must, however, be derived and interpreted carefully. Simplistic and ad hoc approximations are often misleading and could lead to erroneous conclusions. For example, the location of the eigen-values of a linear time-varying system (LTV) does not conclusively determine the system stability in contrast to the linear time-invariant (LTI) case.

3. VARIABLE LINEAR FILTERS

The Kalman-Bucy filter has enjoyed great use in aerospace applications in providing optimal or near-optimal estimates. For its implementation one must solve a linear vector differential equation and a matrix Riccati differential equation. When the signal dynamics, covariances of process and measurement noise, and the observation matrix are known a priori, the solutions of the Riccati equation are pre-computed and the filter gains are stored. There are, however, also estimation problems in which the dynamic equations are time-varying, or the plant noise or measurement noise can be measured but is not known a priori. Such problems, because of the time-varying effects discussed earlier, lead to considerable difficulties in solving them.

The statement of the problem begins with a state vector satisfying the Ito stochastic differential equation

$$dx(t) = F(t)x(t)dt + G(t)d\beta_t \tag{1}$$

and an observation of the form

$$dx(t) = B(t)x(t)dt + d\eta_t \tag{2}$$

As usual x(t) is an n-vector, F(t) is an nxn matrix and G(t) is an nxn matrix and G(t) is an nxr matrix. The Brownian motion β_t is an r-vector with

$$E\left\{d\beta_t \; d\beta_t^T\right\} = Q(t)dt \tag{3}$$

B(t) is an mxn matrix and η_t is an m-vector Brownian motion with

$$E\left\{d\eta_t \; d\eta_t^T\right\} = R(t)dt \tag{4}$$

The minimum variance estimate $\hat{x}$ is given by:

$$d\,\hat{x}\,(t) = F(t)\,\hat{x}\,(t) + P(t)B^T(t)R^{-1}(t)\,[dz(t)-B(t)\hat{x}(t)dt] \tag{5}$$

where P(t) satisfies the Riccati equation

$$\frac{dP}{dt} = F(t)P(t) + P(t)F^T(t) + G(t)Q(t)G^T(t) - P(t)B^T(t)R^{-1}(t)B(t)P(t) \tag{6}$$

and

$$P(t) = E\left\{(x-\hat{x})\;(x-\hat{x})^T\right\} \tag{7}$$

The Riccati matrix differential equation (6) is nonlinear and time-varying and is therefore difficult to solve.

In order to develop approximations it is convenient to convert (6) to the linear form. By the change of variables

$$P = DE^{-1} \tag{8}$$

where D and E are both nxn matrices, eq. (6) can now be written as

$$\begin{pmatrix} \dot{D} \\ \dot{E} \end{pmatrix} = \begin{pmatrix} F & GQG^T \\ B^T R^{-1} B & -F^T \end{pmatrix} \begin{pmatrix} D \\ E \end{pmatrix} \tag{9}$$

With appropriate conditions on the variables D and E compatible with P, we will study the linear equation (9) from the standpoint of approximation theory.

4. APPROXIMATE SOLUTIONS

Equation (9) is of the general form

$$\dot{Y} = M(t)\ Y \tag{10}$$

Of the many kinds of approximations we will be mainly concerned with asymptotic solutions. We consider systems in which the coefficient variation is relatively slow. This is descriptive of a number of aerospace flight vehicle problems. Among the asymptotic techniques we will, in particular, employ the approach of the generalized multiple scales [2].

This technique, although relatively recent in its development, has been applied to a variety of problems in several fields of investigation which have been reported. We will not, therefore, discuss it here in detail. The technique consists of extending the domain of the independent variable, time, into a space of higher dimension by means of "clocks". The differential equations now become partial differential equations in the new space. These are solved asymptotically such that their solutions coincide with the solutions of the original equations along certain lines (called "trajectories") in the extended space. This is tantamount to employing a number of independent observers, each following one aspect of a dynamic phenomenon. A combination of the different — i.e. fast, slow etc.,— behaviors yields the total composite dynamics of the system. Thus a complex phenomenon which cannot be simplified in the original variables can now be described simply in the new time scales.

Several variations of the technique have been developed. For classes of linear time-varying systems they lead to a number of useful results. We will only present a few of these. The interested reader can consult other works on the subject (see references).

I. Scalar Systems

When the linear time-varying (LTV) systems can be decoupled (especially in systems of low order), they lead to equations of the form

$$\sum_{i=0}^{n} \omega_i(t) \frac{d^i y}{dt^i} = f(t) \tag{11}$$

Approximations for slowly varying $\omega_i(t)$ were developed in terms of slow and fast scales (τ_o and τ_1 respectively) by Ramnath [2] as:

$$y(t) \sim \tilde{y}(\tau_o, \tau_1) = \sum_{i=1}^{n} \left[\frac{\partial \phi}{\partial k_i} (\tau_o)\right]^{-\frac{1}{2}} \exp(\tau_{1_i}) \tag{12}$$

where k_i satisfy

$$\phi(k,t) \equiv \sum_{i=o}^{n} \omega_i(t) k^i = o \tag{13}$$

The slow and fast scales are defined by

$$\tau_o = t, \ \tau_{1_i} = \varepsilon^{-1} \int^t k_i(\lambda) d\lambda \tag{14}$$

and ϵ is a small parameter which is a measure of slowness of coefficient variation.

This solution has been used successfully in a number of flight dynamics and control problems dealing with aerospace vehicles. The solution exhibits an asymptotic separation of the rapid and slow behaviors. We note that "clocks" or scale functions are highly nonlinear and can be treated as complex quantities. Special precautions are necessary in dealing with turning point problems.

A variation of this technique is that of developing asymptotic expansions for the clocks themselves. This line of investigation has also been pursued by Ramnath [3]. The approximations are simple and can be simply calculated by paper and pencil procedures. Application of this approach to VTOL aircraft dynamics and parameter sensitivity is discussed in [3]. Further extensions involve more sophisiticated expansions for the clocks.

II. Vector Matrix Formulation 1

Modern control and estimation problems are usually stated in the vector-matrix form. Therefore the required extension of the multiple scales techniques to such forms was developed by Ramnath [4]. This approach can be used for either vector or matrix differential equations and thus for control or estimation problems. In this case the standard form of the system considered is

$$Y = N(t,\varepsilon)\ Y \tag{15}$$

$$= \left(\sum_{i=o}^{n} \varepsilon^i N_i(t)\right) Y \tag{16}$$

where $N_o(t)$ is diagonal. Again, in a region free of turning points, the extension of the generalized multiple scales theory seeks to develop asymptotic approximations of the form

$$\Psi(t,\epsilon) \sim \tilde{\psi}(\tau_i,\varepsilon)\Big|_{\tau_i(t,\varepsilon)} = \alpha(\tau_o,\epsilon)\beta(\underline{\tau}_1(t,\epsilon))\,\gamma(\tau_1(t)) \tag{17}$$

where: $\Psi(t,\varepsilon)$ is the transition matrix of the system (14)

$\tilde{\psi}(\tau_i)$ is the leading approximation for Ψ

$\tau_i(t,\varepsilon)$, $i = -1, 0, 1$ are the scale functions and α, β, γ are the solutions on the different scales. The scale functions $\tau_i(t,\varepsilon)$ are determined systematically in terms of the coefficients $N_i(t)$. They are matrix quantities with elements which are nonlinear and complex. Thus it is a time generalization of the results for the scalar case. This solution recovers the scalar case solution and some well-known classical approximations as special cases.

III. Vector-Matrix Case 2

This is another approach to the approximate analysis of LTV systems in vector or matrix form developed by Ramnath and Bowles [5]. It deals with a system model of the form (10) without the necessity of diagonalizing the coefficient matrix. Again multiple scaling consists of extending time t into a space of higher dimension $\{\tau, T\}$ where $\tau = t$ and $T = \varepsilon K(t)$, a nonlinear matrix quantity. Approximate solutions are developed to the lowest order, in the form:

$$\Psi(t) \sim \alpha(\tau) \exp(T) \tag{18}$$

where T is a quadrature over the coefficient matrix. $\alpha(\tau)$ is determined at the next order. An error analysis was also developed and the approach was applied to the problem of optimal estimation in a missile situation.

5. EXAMPLES

Applications of the generalized multiple scales (GMS) approximations have been made successfully to several aerospace dynamics and control problems. The time-invariant case arises as a natural limit of the GMS solutions. For LTV systems of second order the Liouville-Green (or WKBJ) approximations are readily recovered by the GMS theory. Simpler approximations have been applied to VTOL aircraft problems in [3] and more accurate ones in [6].

Another example (by Ramnath and Bowles [5]) using the matrix multiple scales is that in which a delay tracking loop is trying to maintain a delay lock on a pseudo-random number signal. The delay θ and the frequency ω are taken to satisfy an LTI system of second order. The process and noise covariances q(t) and r(t) are varying. The noise problem is to estimate the time delay and its time rate of change for an electromagnetic signal between a known fixed point and a moving point. The measurement noise $d\eta_t$ comes from both the receiver component noise and from the presence of one or more jammers. The latter can be measured in real time. The extended Kalman filter with a variable $d\eta_t$ will enable us to estimate the time delay which can adapt to changes in measurement noise level $d\eta_t$. It is thus necessary to solve the filter equations very rapidly and therefore GMS approximations were resorted to.

With these considerations application [5] to a system discussed in section 3 is highlighted below.

$$q(t) \sim 10^{-4} \text{ chips}^2/\text{sec}^3 \text{ and } r(t) \sim 10^{-3} \text{ chips}^2/\text{sec} \qquad (19)$$

q(t) is known a priori and r(t) is measured as the mission progresses. The differential equations were also integrated numerically. A comparison shows that the GMS solution is very accurate, being indistinguishable from the numerical one after plotting them both.

Table 1. Comparison of Numerical and
Multiple Scales Solutions at 1 Second

	$P_{\theta\theta}$(x 10^2)	$P_{\theta\omega}$(x 10^3)	$P_{\omega\omega}$(x 10^3)
Multiple Scales	0.670492	0.999801	0.200201
Numerical (0.5 ms)	0.670494	0.9999806	0.200336

A comparison of the approximate computer times for the two solutions shows that the multiple scales approach requires 7 seconds while the numerical integration takes 77 seconds. Thus the GMS solution is much faster for comparable accuracy. This is taken mainly in a qualitative sense. Further refinements are possible with both approaches.

6. SUMMARY AND CONCLUSIONS

Some important results developed by the generalized multiple scales approach for classes of linear time varying systems are presented. They have been successfully applied to a number of dynamics and control problems. However, applications to estimation problems are fewer in number. The GMS solutions show several advantages. They exhibit a separation of fast and slow behaviors and thus contribute to insight. They are accurate and are simply calculable. Therefore computation times are decreased thus opening the door to real time capability. It seems that these advantages warrant the use of such approximations more and more in estimation problems.

REFERENCES

(1) Kokotovic, P., O'Malley, R., Sannuti, P., "Singular Perturbation and Order Reduction in Control Theory - An Overview", Proc. 6th World IFAC Congr. Aug. 1975, paper 51:3.

(2) Ramnath, R.V., Sandri, G., "A Generalized Multiple Scales Approach to a Class of Linear Differential Equations", J.M.A.A. Vol. 29, No. 2, Nov. 1969.

(3) Ramnath, R.V., "Approximate Analysis of Linear Systems with Slowly Varying Coefficients", Report No. 747, Dept. of Aero & Mech. Sci., Princeton University 1965.

(4) Ramnath, R.V., "On the Time Varying Linear Quadratic Problem: A Multiple Scales Approach" Proc. C.C.A.C. - Optimization Days Conference, McGill University, Montreal, Canada 1979. Also, C.S. Draper Laboratory Report No. C4996, 1973.

(5) Ramnath, R.V., Bowles, W.M., "Asymptotic Analysis of a Class of Variable Linear Filters", C.S. Draper Laboratory Report R994 1976.

(6) Ramnath, R.V., "Transition Dynamics of VTOL Aircraft" AIAA Journal, July 1970.

(7) Klimas, A., Ramnath, R.V., Sandri, G., "On The Compatibility Problem in the Uniformization of Asymptotic Expansions", JMAA, Vol. 32, No. 3, December 1970.

AIRCRAFT TRAJECTORY OPTIMIZATION USING SINGULAR PERTURBATION METHODS

RAMAN K. MEHRA and ROBERT B. WASHBURN, JR.

Scientific Systems, Inc., Cambridge, MA 02138

1. ABSTRACT

Nonlinear feedback control laws are derived using Singular Perturbation Methods for the aircraft minimum time long range interception problem assuming complete time scale separation. The theoretical and numerical justification for such separation were presented previously in Refs. [8,9]. This paper describes the details of the application and the nonlinear feedback laws to supplement the material in Refs. [8,9].

2. CALCULATION OF AIRCRAFT FEEDBACK CONTROLS USING SINGULAR PERTURBATION THEORY (SPT)

2.1 Problem Formulation

As we describe in Refs. [6,8,9], SPT seeks to approximate the differential system

$$\begin{aligned} \dot{x} &= f(x,y) \\ \dot{y} &= g(x,y) \end{aligned} \tag{1.1}$$

by constructing an asymptotic approximation (as $\varepsilon \to 0$) to the perturbed system

$$\begin{aligned} \dot{x} &= f(x,y) \\ \varepsilon\dot{y} &= g(x,y) \end{aligned} \tag{1.2}$$

and using this approximation at $\varepsilon=1$ for (1.1). Intuitively, the asymptotic approximation assumes that the y variable in (1.1) varies on a faster time scale than the x variable. Thus, to apply the asymptotic technique successfully, it is necessary to identify the different time scales for the problem and determine which variables are "fast" and which are "slow." Sometimes no separation of time scales is possible for the original variables, and it is necessary to transform to a new set of variables. For nonlinear systems such as the point-mass aircraft model, there is no general method for constructing such a transformation.* In this case, one must rely on physical intuition and exact solutions of the trajectory optimization problem to help identify the fast and slow variables. Since Kaiser [1] it has been recognized that it is useful to replace the velocity state by the energy state in the dynamical equations for a high performance, supersonic aircraft. After this transformation, the energy E and the height h may be separated--E varies on a slower time scale than h. Note that the velocity V and height could not be separated in this way. In terms of energy, the dynamic equations for the point-mass aircraft become

$$\frac{dx}{dt} = V \cos\beta\cos\gamma \tag{1.3}$$

$$\frac{dy}{dt} = V \sin\beta\cos\gamma \tag{1.4}$$

$$\frac{dE}{dt} = \frac{V(uT_{max}\cos\alpha - D)}{mg} \tag{1.5}$$

$$\frac{d\beta}{dt} = \frac{(L + uT_{max}\sin\alpha)\sin\sigma}{mV\cos\gamma} \tag{1.6}$$

$$\frac{dh}{dt} = V \sin\gamma \tag{1.7}$$

$$\frac{d\gamma}{dt} = \frac{(L + uT_{max}\sin\alpha)\cos\sigma - mg\cos\gamma}{mV} \tag{1.8}$$

The constraints are:

$$0 \le \alpha \le \alpha_s \tag{1.9}$$

$$V \le V_{max} \tag{1.10}$$

---

*However, the reader should see [5] for some remarks on finding nonlinear transformations.

$$|\sigma| \le \sigma_{max} \tag{1.11}$$

$$0 \le u \le 1 \tag{1.12}$$

$$0 \le h \le E \tag{1.13}$$

where we let $h_{min} = 0$.

For future reference we indicate the functional form of the aerodynamic terms D, T_{max}, L and V_{max}.

$$D = D_0(h,E) + \eta(h,E)L_\alpha(h,E)\alpha^2 \tag{1.14}$$

$$T_{max} = T_{max}(h,E) \tag{1.15}$$

$$L = L_\alpha(h,E)\alpha \tag{1.16}$$

$$V_{max} = V_{max}(h) \tag{1.17}$$

Note that in the energy state formulation the velocity V is simply shorthand for the function of E and h given by

$$V = \sqrt{2g(E-h)} \tag{1.18}$$

Note also that E is a specific energy measured in units of height. The constraint bounds α_s and σ_{max} are constants in (1.9) and (1.11) respectively. The constraint (1.13) is present only to guarantee that the square root in (1.18) is well-defined. In a model which includes both E and h as state variables this constraint will not be necessary, since the dynamic equations will never allow the height h to exceed the energy E. However, in reduced order models which include only E the constraint will be necessary since h will be a free control.

The control problem is to steer the system (1.3)-(1.8) from an initial state $(x_i,y_i,E_i,\beta_i,h_i,\gamma_i)$ at t_0 to a final state $(x_f,y_f,E_f,\beta_f,h_f,\gamma_f)$ in minimum time t_f-t_0. Thus, the integral cost criterion for this problem is:

$$J(\alpha,\sigma,u) = \int_{t_0}^{t_f} 1 \, dt \tag{1.19}$$

Since the final time t_f is not fixed, the Hamiltonian for this problem is identically 0 along an optimal trajectory. Note that the Hamiltonian is given by

$$H = 1 + \lambda_x V\cos\beta\cos\gamma + \lambda_y V\sin\beta\cos\gamma + \lambda_E\frac{V(uT_{max}\cos\alpha - D)}{mg} + \lambda_\beta\frac{(L+uT_{max}\sin\alpha)\sin\sigma}{mV\cos\gamma} + \lambda_h V\sin\gamma + \lambda_\gamma\frac{(L+uT_{max}\sin\alpha)\cos\sigma - mg\cos\gamma}{mV} \tag{1.20}$$

To approximate the optimal control law for this minimum time problem we will follow the procedure outlined in Ref. [8] of solving a sequence of smaller dimensional control problems corresponding to different time scales for the problem.

At this point we will indicate how the time scale separation was chosen. Previous work such as Refs. [2,3,4,5] and Ref. [7], which considered three dimensional maneuvers, and hence considered turns, indicated that the basic time scale separation consisted of the successively faster groups (x,y), (E,β) and (h,γ). Generally, β was considered faster than E and γ was considered faster than h, although Kelley [5] noted that E might be considered faster than β in some situations. The accuracy of a feedback control calculated with a particular time scale separation depends on the region of state space in which the control is applied. It seems clear from examples that no one time scale separation will be accurate for all regions of state space. Indeed, near the terminal target all time scale separations are inaccurate, i.e., the SPT assumption of any time scale separation is invalid near the terminal target [8].

Although there is as yet no systematic theory to determine the proper time scale separation for a given region of state space, we present the following observations for guidance. The time scale separation for an optimization problem seems to depend on two factors: (1) the sensitivity of the immediate cost with respect to the current state variables and (2) the sensitivity of the long term cost with respect to the current state variables. It appears generally that the most sensitive states should be considered slower than the least sensitive states. In addition, it appears that the sensitivity of the long range cost is important when the target is near (short term problem) and that the sensitivity of the immediate cost is important when the target is far (long range problem). It is important to note that sensitivity analysis of the two kinds of cost, immediate and long-range, can result in different choices of time scales. The right choice depends on the proximity of the target.

The sensitivity of immediate cost is determined mainly by the relative magnitude of the time derivatives, at least in the minimum time optimization problem. The most sensitive states (with respect to immediate cost) are the ones with the smallest derivatives and hence the slowest in terms of actual time variation.

In the following sections we will first work through the analysis required for the solution of each of the control subproblems, and then we will summarize the resulting algorithms and note the computation required for their on-board execution. As we noted in Ref. [8], the SPT approximation involves solving a hierarchy of control problems which must be solved in sequence. Thus, the total time required to calculate one update of the aircraft controls by means of the SPT method is the sum of the computation times required for each of the algorithms which solve the individual subproblems. In addition to noting the computation time required for the individual subproblem algorithms, we will also note which functions must be stored and which quantities could be obtained directly from aircraft sensors.

The organization of this section roughly follows the separation of time scales we have assumed for the aircraft model. Thus, Section 2.2 discusses the x,y reduced order problem, Section 2.3 discusses the E boundary layer problem, and Section 2.4 discusses the β boundary layer problem. Then in Section 2.5 we discuss the h boundary layer problem and finally, in Section 2.6 we discuss the γ boundary layer problem which operates on the fastest time scale.

Figure 1 shows the complete control logic for long range interception. It involves an iterative calculation of the intercept point and the intercept time, t_f. In addition, the calculation of heading on the cruise arc and controls for the three parts of the trajectory (before cruise, cruise and after cruise) is also performed. These segments of the optimal trajectory are defined in the next section.

2.2 The Reduced Order Problem in x,y: The Cruise Solution

2.2.1 x,y Reduced Order Control Problem and Feedback Solution. By considering x and y on the same time scale and all the other states in (1.3)-(1.8) on a faster time scale we obtain the reduced order dynamical system

$$\frac{dx}{dt} = V \cos\beta\cos\gamma \tag{2.1}$$

$$\frac{dy}{dt} = V \sin\beta\cos\gamma \tag{2.2}$$

$$0 = \frac{V(uT_{max}\cos\alpha - D)}{mg} \tag{2.3}$$

$$0 = \frac{(L + uT_{max}\sin\alpha)\sin\sigma}{mV\cos\gamma} \tag{2.4}$$

$$0 = V\sin\gamma \tag{2.5}$$

$$0 = \frac{(L + uT_{max}\sin\alpha)\cos\sigma - mg\cos\gamma}{mV} \tag{2.6}$$

together with the constraints (1.9)-(1.13). The Hamiltonian for this problem is

$$H = 1 + \lambda_x V \cos\beta\cos\gamma + \lambda_y V \sin\beta\cos\gamma \tag{2.7}$$

and since the problem is to minimize time, the Hamiltonian is identically 0 along the optimal trajectory. The pseudocontrols E, h, β, γ and the controls α, σ, u are chosen to maximize the velocity V subject to the equality and inequality constraints and satisfy the x, y terminal conditions. Thus, one easily finds that the optimal pseudocontrols for the reduced order problem are given by:

$$E^* = E_c \tag{2.8}$$

$$\beta^* = \tan^{-1} \frac{y_f - y_i}{x_f - x_i} \tag{2.9}$$

$$h^* = h_c \tag{2.10}$$

$$\gamma^* = 0 \tag{2.11}$$

$$\alpha^* = \alpha_c \tag{2.12}$$

$$\sigma^* = 0 \tag{2.13}$$

$$u^* = u_c \tag{2.14}$$

where the subscript "c" denotes constant cruise values. Let V_c denote the corresponding maximum cruise velocity determined from E_c and h_c by (1.18). For our particular aircraft, which is taken from [7], these constants are given by

$$E_c = 2.9949 \times 10^4 \text{ m} \tag{2.15}$$

$$h_c = 1.2192 \times 10^4 \text{ m} \tag{2.16}$$

$$\alpha_c = 1.509^\circ \tag{2.17}$$

$$u_c = .8928 \tag{2.18}$$

$$V_c = 590.2 \text{ ms}^{-1}$$

In order to define the cost criterion for the E boundary layer we need to obtain the adjoints λ_x and λ_y for this reduced order problem. This is easy to do and we find that they are given by the expressions

$$\lambda_x = -\frac{\cos\beta_c}{V_c} \tag{2.20}$$

$$\lambda_y = -\frac{\sin\beta_c}{V_c} \tag{2.21}$$

where we have introduced the notation β_c for the angle given by (2.9) or by

$$\cos\beta_c = \frac{x_f - x_i}{((x_f - x_i)^2 + (y_f - y_i)^2)^{\frac{1}{2}}} \tag{2.22}$$

$$\sin\beta_c = \frac{y_f - y_i}{((x_f - x_i)^2 + (y_f - y_i)^2)^{\frac{1}{2}}} \tag{2.23}$$

Note that (2.22) and (2.23) uniquely define β_c as the heading angle between the initial point (x_i, y_i) and the final point (x_f, y_f), measured from the positive x direction to the line between these points. Note that we can rewrite the control law for β^* in (2.9) as

$$\beta^* = \beta_c \tag{2.24}$$

For convenience we will use the simpler expression (2.24) in the future rather than the expression (2.9). However, it should be kept in mind that β_c is a function of the current x and y states as expressed in (2.9) or in (2.22), (2.23).

The cost criterion for the next level of the SPT hierarchy of control subproblems is found by substituting the expressions for the adjoints λ_x and

λ_y from (2.20), (2.21) into (2.7) to obtain

$$\hat{L} = 1 - \frac{V}{V_c} \cos\gamma \cos(\beta - \beta_c) \tag{2.25}$$

2.2.2 Linearization Around the x,y Reduced Order Solution. Exploiting the possibility of linearizing the boundary layer problem around the reduced order solution, we obtain a linear, quadratic criterion, time invariant, infinite time problem in the perturbations of E, β, h, γ, u, σ and α from their optimal values computed in the x,y reduced order solution. The linear control problem has a simple control law of the form

$$\begin{pmatrix} u \\ \sigma \\ \alpha \end{pmatrix} = \begin{pmatrix} u_c \\ 0 \\ \alpha_c \end{pmatrix} + G \begin{pmatrix} E-E_c \\ \beta-\beta_c \\ h-h_c \\ \gamma \end{pmatrix} \tag{2.26}$$

where G is a constant 3 × 4 gain matrix computed off-line.

The equation (2.26) gives us the linear regulator control to maintain the cruise conditions of the aircraft. Thus, on the cruise portion of the trajectory the full control law for the aircraft is particularly simple.

2.3 E-Boundary Layer Problem and Feedback Solution

By considering E on a faster time scale than x or y and on a slower time scale than β, h or γ, we obtain the following boundary layer dynamical system for E:

$$\frac{dE}{dt} = \frac{V(uT_{max}\cos\alpha - D)}{mg} \tag{3.1}$$

$$0 = \frac{(L + uT_{max}\sin\alpha)\sin\sigma}{mV\cos\gamma} \tag{3.2}$$

$$0 = V\sin\gamma \tag{3.3}$$

$$0 = \frac{(L + uT_{max}\sin\alpha)\cos\sigma - mg\cos\gamma}{mV}$$

together with the constraints (1.9)-(1.13). The cost criterion for this problem is found from (2.25) to be

$$J = \int_0^\infty \left[1 - \frac{V}{V_c}\cos\gamma\cos(\beta-\beta_c)\right] d\tau \tag{3.5}$$

The initial condition is given by $E = E_i$ but the final condition is given by $E = E_c$ and not by $E = E_f$. In particular, we see that the resulting SPT control law will fly the plane to the cruise condition but will not yield a control which will give some final energy other than the cruise energy.

Since the control problem for this boundary layer has only one state dimension, we may use the technique of Ref. [6,8] to solve for the feedback control law by minimizing or maximizing the expression

$$\frac{mg\left(1 - \frac{V}{V_c}\cos\gamma\cos(\beta-\beta_c)\right)}{V(uT_{max}\cos\alpha - D)} \tag{3.6}$$

with respect to the pseudocontrols β, h, γ and the controls u, σ, α such that the inequality constraints of (1.9)-(1.13) and the equality constraints of (3.2)-(3.4) are maintained. In addition, if we are minimizing (3.6), then we must restrict ourselves to pseudocontrol and control values such that $\frac{dE}{dt}$ given by (3.1) is positive, and if we are maximizing (3.6), then we must restrict ourselves such that $\frac{dE}{dt}$ is negative.

From the equality constraints it is not hard to deduce that

$$\gamma = 0 \tag{3.7}$$

$$\sigma = 0 \tag{3.8}$$

Thus, the pseudocontrol γ and the control σ are unchanged from their values given by the x,y reduced order solution. Moreover, it is not hard to see that the optimization problem gives

$$\beta = \beta_c \tag{3.9}$$

so that the pseudocontrol β also has its reduced order value. Substituting (3.7), (3.8) and (3.9) into the original E-boundary layer problem, we obtain the following optimization problem. We must maximize or minimize the function

$$\frac{mg(1 - \frac{V}{V_c})}{V(uT_{max} \cos\alpha - D)} \tag{3.10}$$

subject to the equality constraint

$$0 = L + uT_{max} \sin\alpha - mg \tag{3.11}$$

together with the inequality constraints (1.9), (1.10), (1.11), (1.12). Using the equality constraint (3.11) to solve for u and substituting this expression back into (3.10) we obtain the expression

$$\frac{mg(1 - \frac{V}{V_c})}{V([mg - L]\cot\alpha - D)} \tag{3.12}$$

which we must maximize or minimize, according to the convention described above after (3.6). By eliminating u we also obtain a new inequality constraint on V and α corresponding to the original constraint on u in (1.12). The new constraint is

$$0 \le \frac{mg - L}{T_{max} \sin\alpha} \le 1 \tag{3.13}$$

We have now reduced the optimization problem to a problem involving only two variables, V and α. Let us now consider V fixed and examine the dependence of (3.12) on α. Using (1.14)-(1.16) we can write the α dependence explicitly in (3.12) as

$$\frac{mg(1 - \frac{V}{V_c})}{V([mg - L_\alpha \alpha]\cot\alpha - D_0 - L_\alpha \eta\alpha^2)} \tag{3.14}$$

Similarly, the inequality constraint (3.13) becomes

$$0 \le \frac{mg - L_\alpha \alpha}{T_{max} \sin\alpha} \le 1 \tag{3.15}$$

It is not hard to see that for α in the range $0 < \alpha < \pi$ the denominator of (3.14), namely

$$V([mg - L_\alpha \alpha]\cot\alpha - D_0 - L_\alpha \eta\alpha^2) \tag{3.16}$$

is a strictly decreasing function of α. Thus, the expression (3.14) is an increasing function of α in this range. Note that this range easily includes all admissible α since $\alpha_s < \pi$ (in our aircraft example $\alpha_s = 12°$).

Since (3.14) is monotonically increasing in α, in order to maximize (3.14) with respect to α we must choose the largest possible α consistent with the inequality constraints (3.15) and (1.9). Similarly, if we wish to minimize (3.14), we must choose the smallest possible α consistent with the inequality constraints.

Note that the function

$$u(\alpha) = \frac{mg - L_\alpha \alpha}{T_{max} \sin\alpha} \tag{3.17}$$

which appears in the inequality constraint (3.15) is monotonically decreasing in α for admissible α (in fact, for all α such that $0 < \alpha < \pi/2$). The monotonic behavior of the function (3.17) allows us to describe the maximum and minimum allowable values of α fairly easily. There are essentially three different cases to consider.

Case 1. This case corresponds to

$$L_\alpha \alpha_s + T_{max} \sin\alpha_s < mg \tag{3.18}$$

In this case there is no admissible α which satisfies the inequality constraints (3.15) and (1.9). Physically, (3.18) represents the stall constraint for the plane and it defines the region in which the plane cannot maintain vertical equilibrium using both lift and thrust forces.

Case 2. This case corresponds to the two relations

$$L_\alpha \alpha_s < mg \tag{3.19}$$

and

$$T_{max} \sin\alpha_s + L_\alpha \alpha_s \geq mg \tag{3.20}$$

Physically, there is less lift force than the weight of the aircraft, but by using maximum thrust in addition to lift, the plane is able to maintain equilibrium. The maximum admissible α is given by $\alpha = \alpha_s$. The minimum admissible α is given by solving the equation

$$T_{max} \sin\alpha + L_\alpha \alpha = mg \tag{3.21}$$

Case 3. This case corresponds to

$$L_\alpha \alpha_s \geq mg \tag{3.22}$$

Physically, the aircraft has enough lift to maintain vertical equilibrium by lift force alone. The maximum admissible α is given by

$$\alpha = \frac{mg}{L_\alpha} \tag{3.23}$$

and the minimum admissible α is given by solving (3.21) as above.

The argument above concerning α allows us to express α as a function of V and E, and hence allows us to eliminate α from the problem to optimize (3.14). Since the aerodynamic quantities L_α, D_0 and T_{max} are given only as data, we must solve the V optimization numerically. We have done this off-line and stored the resulting optimal control laws V*(E). Note that there are two V* controls, one corresponding to the maximization and one corresponding to the minimization of (3.14). We have found that for the optimal $V = V^*(E)$, only Case 3 above occurs. Thus, the throttle control u is 0 or 1 depending on whether E is above or below the cruise energy E_c. The angle-of-attack α is chosen to maintain vertical equilibrium, i.e., to satisfy equation (3.23) if $u = 0$ and to satisfy (3.21) if $u = 1$.

To obtain the cost criterion for the next level control problem, we need to obtain the optimal adjoint λ_E^* for the E boundary layer problem. This is easily accomplished by solving

$$0 = 1 - \frac{V^*}{V_c} + \lambda_E^* \frac{V^*}{mg} (T^* \cos\alpha^* - D^*) \tag{3.24}$$

Note that (3.24) gives λ_E^* as a function of V*, which in turn is a function of E, namely

$$\lambda_E^* = \frac{mg(\frac{V^*}{V_c} - 1)}{V^*(T^* \cos\alpha^* - D^*)} \tag{3.25}$$

We may either store λ_E^* explicitly as a function of E or we may compute λ_E^* from V*(E) using (3.25). However, the latter approach requires us to store the aerodynamic functions T*,D*.

2.4 β-Boundary Layer Problem and Feedback Solution

By considering β on a faster time scale than x, y or E and on a slower time scale than h or γ, we obtain the following boundary layer dynamical system for β:

$$\frac{d\beta}{dt} = \frac{(L + uT_{max} \sin\alpha)\sin\sigma}{mV \cos\gamma} \tag{4.1}$$

$$0 = V \sin\gamma \tag{4.2}$$

$$0 = \frac{(L + uT_{max} \sin\alpha)\cos\sigma - mg \cos\gamma}{mV} \tag{4.3}$$

together with the constraints (1.9)-(1.13). The cost criterion for this problem is found to be

$$J = \int_0^\infty [1 - \frac{V}{V_c} \cos\gamma \cos(\beta-\beta_c) + \lambda_E^* \frac{V}{mg} (uT_{max} \cos\alpha - D)]\, d\tau \tag{4.4}$$

where λ_E^* is the optimal adjoint computed in (3.25). The initial condition is given by $\beta = \beta_i$ but the final condition is $\beta = \beta_c$. Again, as for the E-boundary layer problem, we see that the SPT control law will fly the plane to the cruise heading but will not yield a control which will give the final heading angle.

The optimal control law is obtained by minimizing or maximizing the expression

$$\frac{[1 - \frac{V}{V_c} \cos\gamma \cos(\beta-\beta_c) + \lambda_E^* \frac{V}{mg} (uT_{max} \cos\alpha - D)]mV \cos\gamma}{(L + uT_{max} \sin\alpha)\sin\sigma} \tag{4.5}$$

with respect to the pseudocontrols h, γ and the controls u, σ, α such that the inequality constraints of (1.9)-(1.13) and the equality constraints (4.2) and (4.3) are satisfied. If $\beta < \beta_c$ then we minimize (4.5) and we must restrict ourselves to pseudocontrol and control values which give a positive value for $d\beta/dt$ in (4.1). Likewise, if $\beta > \beta_c$ then we maximize (4.5) and we must restrict ourselves to pseudocontrols and controls such that $d\beta/dt$ remains negative.

From the equality constraint (4.2) it is not hard to see that

$$\gamma = 0 \tag{4.6}$$

At this stage we will make the additional assumption that u takes on the same value (namely 1 or 0) that it does in the E-boundary layer solution. In general, we assume that if a control variable takes only a maximum or minimum value at a slower level of the SPT hierarchy, then it maintains that value of the control at the faster levels of the hierarchy. Thus, instead of writing uT_{max} we will write T and assume that u is already determined and fixed by the E-boundary layer solution.*

With these assumptions the expression (4.3) becomes

$$0 = (L + T \sin\alpha)\cos\sigma - mg \tag{4.7}$$

Using (4.7) to simplify the denominator of (4.5), we obtain

$$\frac{[1 - \frac{V}{V_c} \cos(\beta-\beta_c) + \lambda_E^* \frac{V}{mg} (T \cos\alpha - D)]V}{g \tan\sigma} \tag{4.8}$$

Let us fix V and try to determine the optimum α and σ for a given fixed V. The result is:

$$\tan^2\sigma = \frac{1 - \frac{V}{V_c} \cos(\beta-\beta_c) + \lambda_E^* \frac{V}{mg} (T - D_0)}{-\lambda_E^* V \frac{(T + 2\eta L_\alpha)}{(T + L_\alpha)^2} mg} \tag{4.9}$$

In (4.9) one takes the positive or negative root depending on whether $\beta < \beta_c$ or $\beta > \beta_c$. The corresponding value for α is determined from (4.7) using the value for σ from (4.9). Note that we must also test the possibility that σ or α lie on a constraint. One does this simply by setting $\sigma = \pm\sigma_{max}$ and solving (4.7) for the corresponding α or by setting $\alpha = \alpha_s$ and solving (4.7) for

---

*A comparison with the exact energy state solution using both E and β as states on the same time scale shows that this approximation is valid in all cases except those where level flight cannot be maintained during a turn with zero thrust. This special case is handled quite easily in practice by switching to full thrust if level flight cannot be maintained otherwise.

σ. We then test these values against the value of σ determined by (4.9) and keep the one which gives the smallest (if $\beta < \beta_c$) or greatest (if $\beta > \beta_c$) value of (4.8).

So far we have only determined the optimum σ and α values in terms of a fixed V. To obtain the optimum V, and thus solve the problem, we resort to numerical optimization off-line. This optimization is made difficult by the dependence of (4.8) on E and $\beta-\beta_c$. Thus, the computed optimal velocity pseudocontrol V* for the β-boundary layer is a function of the following form

$$V^* = V^*(E,\beta-\beta_c) \tag{4.10}$$

Note that $V^*(E,\beta-\beta_c) = V^*(E,-(\beta-\beta_c))$. Another control law based on the exact (E,β) solution of Ref. [7] is given in Ref. [6].

2.5 h-Boundary Layer Problem and Feedback Solution

As discussed previously, we have separated the vertical plane part of the trajectory from the horizontal turn part. Thus, in considering the h-boundary layer we need only look at the vertical plane case with $\sigma = 0$ and $\beta = \beta_c$. Moreover, we assume that any controls which take their value on a constraint in slower levels retain the constraint value at faster levels.* Consequently, we assume that T is determined ($u = 0$ if $E > E_c$ and $u = 1$ if $E < E_c$). With these assumptions the dynamical system for the h boundary layer becomes

$$\frac{dh}{d\tau} = V \sin\gamma \tag{5.1}$$

with the equality constraint

$$0 = (L + T \sin\alpha) - mg \cos\gamma \tag{5.2}$$

and the cost criterion

$$J = \int_0^\infty \left[1 - \frac{V}{V_c}\cos\gamma + \lambda_E^* \frac{V}{mg}(T\cos\alpha - D)\right] d\tau \tag{5.3}$$

where λ_E^* is the optimal adjoint computed in (3.25). The initial condition is $h = h_i$ and the final condition is $h = h^*(E)$, where h^* is determined from E and V* using Eq. (1.18).

The optimal control law is obtained by minimizing or maximizing the expression

$$\frac{\left[1 - \frac{V}{V_c}\cos\gamma + \lambda_E^* \frac{V}{mg}(T\cos\alpha - D_0 - \eta L_\alpha \alpha^2)\right]}{V \sin\gamma} \tag{5.4}$$

with respect to the pseudocontrol γ and the control α such that the equality constraint (5.2) is satisfied. If $h < h^*(E)$, then we minimize (5.4) and we must restrict ourselves to $\gamma > 0$. If $h > h^*(E)$, then we maximize (5.4) and we must restrict ourselves to $\gamma < 0$. Note that the expression (5.4) is an odd function of γ and that the expression (5.2) is an even function of γ. It follows that the minimum solution $\gamma > 0$ is the negative of the maximum solution $\gamma < 0$ and that α is the same in both cases. Thus, we restrict our attention to the case when $\gamma > 0$.

To determine the solution $\gamma > 0$ that minimizes the expression (5.4), define the Lagrangian function $\mathscr{L}$ as

$$\mathscr{L} = \frac{\left[1 - \frac{V}{V_c}\cos\gamma + \lambda_E^* \frac{V}{mg}(T\cos\alpha - D_0 - \eta L_\alpha \alpha^2)\right]}{V \sin\gamma} + \frac{\lambda[L_\alpha \alpha + T\sin\alpha - mg\cos\gamma]}{V\sin\gamma} \tag{5.5}$$

where λ is an undetermined Lagrange multiplier. Setting $\partial\mathscr{L}/\partial\gamma = \partial\mathscr{L}/\partial\alpha = 0$, we obtain the relations

$$0 = \left(\frac{V}{V_c} + \lambda mg\right) - \left(1 + \lambda_E^* \frac{V}{mg}(T\cos\alpha - D_0 - \alpha^2\eta L_\alpha) + \lambda(L_\alpha\alpha + T\sin\alpha)\right)\cos\gamma \tag{5.6}$$

---

*More general solutions without these assumptions are given in Ref. [6]. In particular, it is shown that, for turning maneuvers with $\sigma \neq 0$, the results of Sections 2.5 and 2.6 are extended quite easily.

$$0 = -\lambda_E^* \frac{V}{mg} (T \sin\alpha + 2\eta L_\alpha \alpha) + \lambda(L_\alpha + T \cos\alpha) \qquad (5.7)$$

Assuming that α is small enough to neglect in (5.6), (5.7),* we find that λmg in (5.6) is negligible and we obtain the following expression for $\cos\gamma$

$$\frac{\frac{V}{V_c}}{1 + \lambda_E^* \frac{V}{mg} (T - D_0)} = \cos\gamma \qquad (5.8)$$

We solve (5.8) for $\cos\gamma$ and then obtain α from (5.2). Note that for values of h close to h*, the expression (5.8) may exceed 1 slightly (by .02) due to the absence of the nonzero α in the denominator. In this case we can either set $\gamma^* = 0$, since it will be very close to 0, or we can use a method of successive approximation to obtain a better approximation--for example, solve (5.2) with $\gamma = 0$ and use this approximation of α in (5.6), (5.7) to obtain a better approximation of γ. In Ref. [6] we have carried out a more careful analysis of the solution of equations (5.2), (5.6), (5.7) using perturbation analysis.

The adjoint λ_h^* for the h-boundary layer is obtained from solving

$$0 = 1 - \frac{V}{V_c} \cos\gamma + \lambda_E^* \frac{V}{mg} (T \cos\alpha - D_0 - \eta L_\alpha \alpha^2) + \lambda_h^* V \sin\gamma \qquad (5.9)$$

for λ_h^*, where γ and α take their optimum values for the h-boundary layer problem.

2.6 γ-Boundary Layer Problem and Feedback Solution

We make the same assumptions for this problem that we did for the h-boundary layer. In particular, we assume that T is fixed, and $\sigma = 0$ and $\beta = \beta_c$. The dynamic system for the γ-boundary layer becomes

$$\frac{d\gamma}{d\tau} = L_\alpha \alpha + T \sin\alpha - mg \cos\gamma \qquad (6.1)$$

with the cost criterion

$$J = \int_0^\infty [1 - \frac{V}{V_c} \cos\gamma + \lambda_E^* \frac{V}{mg} (T \cos\alpha - D) + \lambda_h^* V \sin\gamma]\, d\tau \qquad (6.2)$$

The optimal control law is obtained by minimizing or maximizing the expression

$$\frac{1 - \frac{V}{V_c} \cos\gamma + \lambda_E^* \frac{V}{mg} (T \cos\alpha - D_0 - \eta L_\alpha \alpha^2) + \lambda_h^* V \sin\gamma}{L_\alpha \alpha + T \sin\alpha - mg \cos\gamma} \qquad (6.3)$$

with respect to the control α. If $\gamma < \gamma^*(E,h)$, then we minimize (6.3) and we must restrict ourselves to α such that $L_\alpha \alpha + T \sin\alpha - mg \cos\gamma > 0$. If $\gamma > \gamma^*(E,h)$ then we must maximize (6.3) and we must restrict ourselves to α such that $L_\alpha \alpha + T \sin\alpha - mg \cos\gamma < 0$.

Taking the derivative of (6.3) gives us

$$\{-(L_\alpha \alpha + T \sin\alpha - mg \cos\gamma)(\lambda_E^* \frac{V}{mg} [T \sin\alpha + 2\eta L_\alpha \alpha]) \qquad (6.4)$$

$$-(L_\alpha + T \cos\alpha)(1 - \frac{V}{V_c} \cos\gamma + \lambda_E^* \frac{V}{mg} (T \cos\alpha - D) + \lambda_h^* V \sin\gamma)\}(L_\alpha \alpha + T \sin\alpha - mg \cos\gamma)^{-2}$$

When either $E > E_c$ and $\gamma < \gamma^*$ or when $E < E_c$ and $\gamma > \gamma^*$, the expression (6.4) is negative. Hence, in these cases the expression (6.3) is decreasing in α. When $E < E_c$ and $\gamma > \gamma^*$, we want to maximize (6.3) so we choose $\alpha = 0$. When $E > E_c$ and $\gamma < \gamma^*$, we want to minimize (6.3) so we choose $\alpha = \alpha_s$.

To solve for α in the other cases, we set (6.4) equal to 0 and we neglect terms in α higher than linear. Define α_0 by

$$\alpha_0 = \frac{mg \cos}{L_\alpha + T} \qquad (6.5)$$

---

*This approximation is quite good for $T = T_{max}$, but needs refinement for $T=0$.

Then setting (6.4) equal to 0 gives the following quadratic equation,

$$\alpha^2 - \alpha_0\alpha + k = 0 \tag{6.6}$$

where k is given by

$$k = \frac{1 - \frac{V}{V_c}\cos\gamma + \lambda_E^* \frac{V}{mg}(T - D_0) + \lambda_h^* V \sin\gamma}{\lambda_E^* \frac{V}{mg}(T + 2\eta L_\alpha)} \tag{6.7}$$

Note that $k < 0$ for $E < E_c$ and $k > 0$ for $E > E_c$. The solution α is given by

$$\alpha = \frac{\alpha_0}{2} + \tfrac{1}{2}\sqrt{\alpha_0^2 - 4k} \tag{6.8}$$

when $E < E_c$, and

$$\alpha = \frac{\alpha_0}{2} - \tfrac{1}{2}\sqrt{\alpha_0^2 - 4k} \tag{6.9}$$

when $E > E_c$. The complete hierarchical nonlinear feedback control law resulting from the above SPT calculations is shown in Fig. 2.

3. CONCLUSION

It has been shown how the aircraft long range minimum time intercept problem can be solved by using Singular Parturbation Theory (SPT). The resulting algorithms are in a feedback form and can be implemented on a microcomputer for on-line trajectory optimization. Further details of such an implementation and the numerical simulation results are contained in Ref. [6].

4. REFERENCES

1. Kaiser, A. (1944), "The Climb of Jet-Propelled Aircraft--Part 1. Speed Along the Path in Optimum Climb," Ministry of Supply RTP/TlB, Translation GDC/15/148T.
2. Kelley, H. J. (1971a), "Flight Path Optimization with Multiple Time Scales," J. Aircraft, Vol. 8, p. 238.
3. Kelley, H. J. (1971b), "Reduced-Order Modeling in Aircraft Mission Analysis," AIAA J., Vol. 9, p. 349.
4. Kelley, H. J. and L. Lefton (1972), "Supersonic Aircraft Energy Turns," 5th IFAC Congress, Paris; also Automatica, Vol. 8, p. 575.
5. Kelley, H. J. (1973), "Aircraft Maneuver Optimization by Reduced Order Approximation," in Control and Dynamic Systems (ed. C. T. Leondes), New York: Academic Press.
6. Mehra, R. K. et al. (1979), "A Study of the Application of Singular Perturbation Theory," NASA Contractor Report 3167.
7. Parsons, M. G. and A. E. Bryson (1972), "Three-Dimensional, Minimum-Time Flight Paths to a Point and Onto a Line for a Supersonic Aircraft with a Maximum Mach Number Constraint," Stanford Univ., Center for Systems Research, Report SUDAAR No. 444.
8. Washburn, R. B. et al. (1978), "Application of Singular Perturbation Techniques (SPT) and Continuation Methods for On-Line Aircraft Trajectory Optimization," Proc. IEEE Conf. on Dec. and Cont., pp. 983-990.
9. Washburn, R. B. and R. K. Mehra (1979), "Asymptotic Continuation Method for Trajectory Optimization," Proc. Joint Auto. Cont. Conf., p.617-621.

ACKNOWLEDGEMENT

The work reported here was supported by NASA Langley Research Center under Contract NAS1-15113.

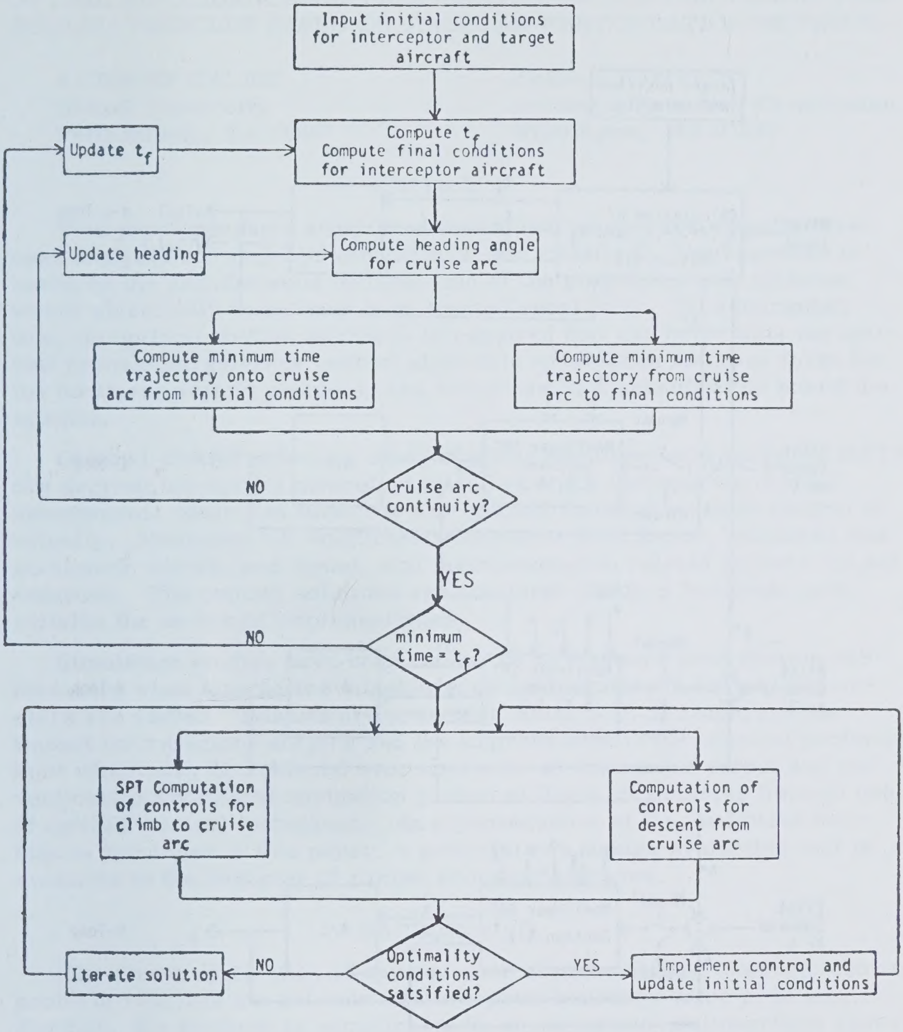

Input initial conditions for interceptor and target aircraft
Update t_f
Compute t_f Compute final conditions for interceptor aircraft
Update heading
Compute heading angle for cruise arc
Compute minimum time trajectory onto cruise arc from initial conditions
Compute minimum time trajectory from cruise arc to final conditions
NO
Cruise arc continuity?
YES
NO
minimum time = t_f?
SPT Computation of controls for climb to cruise arc
Computation of controls for descent from cruise arc
Iterate solution
NO
Optimality conditions satsified?
YES
Implement control and update initial conditions

Figure 1

Overall Control Logic for Long Range Interception

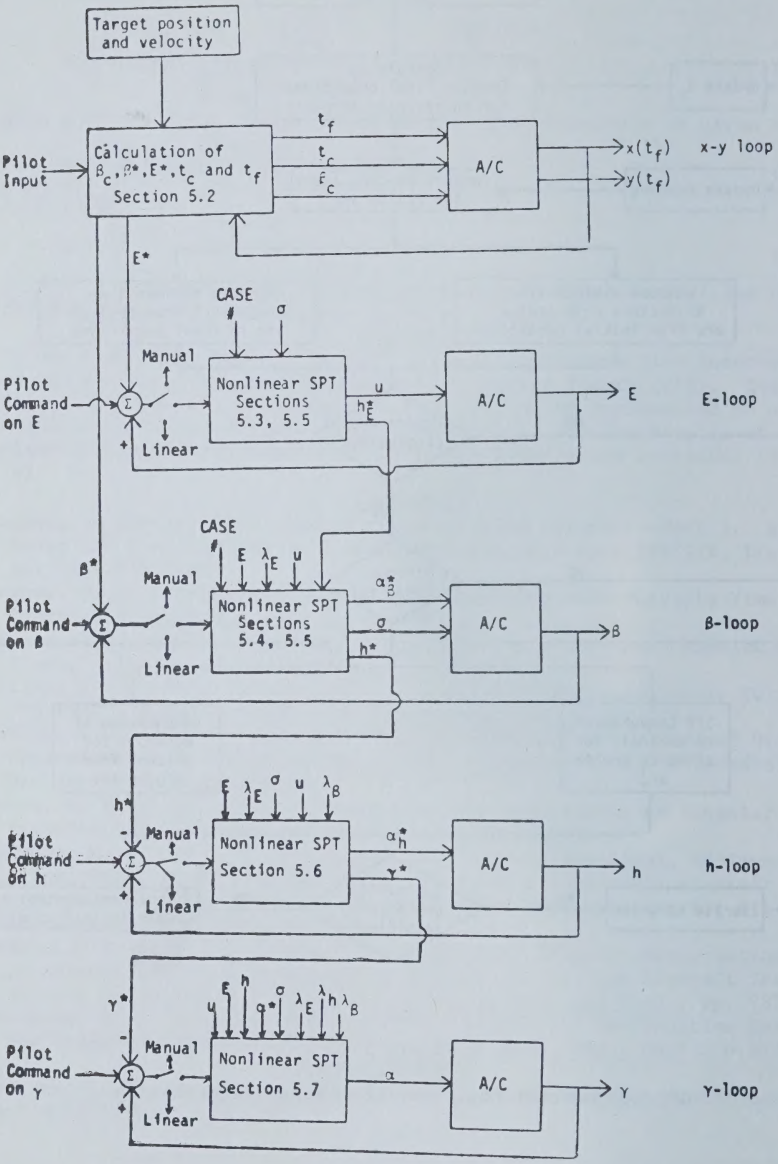

Target position and velocity
Pilot Input
Calculation of β_c, β^*, E^*, t_c and t_f Section 5.2
t_f
t_c
ε_c
A/C
$x(t_f)$
$y(t_f)$
x-y loop
E^*
CASE #
σ
Manual
Pilot Command on E
Σ
+
Linear
Nonlinear SPT Sections 5.3, 5.5
u
h^*_E
A/C
E
E-loop
CASE #
E
λ_E
u
β^*
Manual
Pilot Command on β
Σ
Linear
Nonlinear SPT Sections 5.4, 5.5
α^*_β
σ
h^*
A/C
β
β-loop
E
λ_E
σ
u
λ_β
h^*
-
Manual
Pilot Command on h
Σ
+
Linear
Nonlinear SPT Section 5.6
α^*_h
γ^*
A/C
h
h-loop
u
E
h
α^*
σ
λ_E
λ_h
λ_β
γ^*
-
Manual
Pilot Command on γ
Σ
+
Linear
Nonlinear SPT Section 5.7
α
A/C
γ
γ-loop

Figure 2

Hierarchical Feedback Structure of the On-Line SPT Algorithm

OPTIMIZED GUIDANCE AND PROPULSION CONTROL FOR RAMJET PROPELLED VEHICLES BASED ON SINGULAR PERTURBATION METHODS

ANTHONY CALISE
Drexel University
Philadelphia, PA 19087

FRED GOLDSTEIN
Dynamics Research Corporation
Wilmington, MA 01887

ABSTRACT

This paper presents an optimal component integration and effectiveness analysis of ramjet powered air-to-air missiles. The approach is based on the simultaneous optimization of the propulsion and guidance, where classically these have been treated separately. To accomplish this, an optimal control approach is required that can determine the optimal propulsion/guidance control algorithm which is in feedback form for the nonlinear ramjet problem, and which can be mechanized on board the missile.

Optimal control solutions using singular perturbation techniques (SPT) are derived for ramjet controlled missiles which incorporate optimal aerodynamic control of turn rate and optimal thrust magnitude control of velocity. Maximum lift coefficient, maximum load factor, minimum and maximum velocity and thrust, and instrumentation related constraints are enforced. The control solutions are nonlinear and in a feedback form suitable for on-board implementation.

Simulation studies have been conducted to measure total system performance when alternative mission/propulsion/guidance related parameters are varied. Results are presented which both demonstrate the impact on trajectory shaping and the improvement in engagement performance which may be achieved over conventional (maximum thrust and conventional proportional navigation guidance) flight trajectories through use of optimal control techniques. As a consequence of the analytical techniques developed in this paper, a powerful new design evaluation tool is available to the designer of ramjet propelled vehicles.

BACKGROUND

The direct application of optimal control theory to the intercept control problem requires the solution of a two-point boundary value problem. Further, the problem is complicated by aerodynamic nonlinearities associated with lift, thrust and drag, and control constraints forced by structural and angle of attack limits. These complications result in large computation time so that the resulting solution is unsuitable for both missile performance evaluation and implementation on an airborne computer.

In the presence of these formidable difficulties, the approach taken here is to obtain a near optimal solution to the problem based on singular perturbation theory [1, 2, 3]. This method provides an inherently analytic solution to the optimal control problem that satisfies all optimality conditions, while enforcing a broad class of state and control constraints. This

---

Research sponsored by Air Force Aero Propulsion Lab, (RJT), Wright-Patterson AFB, Ohio, under Contract No. F66315-77-C-2072.

is achieved by reducing the original problem to a series of lower order problems that are easily solved. Model order reduction is achieved by initially ignoring the faster (transient) dynamics. These are later accounted for in a separate boundary layer analysis. The theory insures that the composite solution (outer plus boundary layer) provides a good approximation to the original problem. The resulting control law is in feedback form and due to its analytic form is suitable for implementation in an onboard computer.

There are several distinguishing features of this study as compared to previous research. Ref. [4] considers flight path optimization for transport aircraft. Ref. [5] develops the optimal three dimensional control for rocket powered missiles based on a velocity state approach. While this approach is valid for the rocket powered intercept control problem, the lower thrust/weight ratios of ramjet powered vehicles requires an alternative form of modeling [6]. This paper will present an energy state modeling approach to derive the optimal control laws.

FORMULATION

Formulation of the optimal control solution to the intercept control problem is based on singular perturbation theory (SPT). The equations of motion are written in a coordinate frame shown in Fig. 1. The y_r axis is along the horizontal component of the target velocity vector, and x_r defines cross-range relative to target. The equations of motion are:

$$\dot{x}_r = v \cos \gamma \cos \phi \tag{1}$$

$$\dot{y}_r = v \cos \gamma \sin \phi - v_t \cos \gamma_t \tag{2}$$

$$\epsilon \dot{E} = (T - D)\, v/w \tag{3}$$

$$\epsilon^2 \dot{z} = v \sin \gamma \tag{4}$$

$$\epsilon^2 \dot{\phi} = g\, L_2/w\, v \cos \gamma \tag{5}$$

$$\epsilon^2 \dot{\gamma} = g\, [L_1/w - \cos \gamma]/v \tag{6}$$

Singular perturbation methods can be used to effect a time scale separation of the system states. Thus, the system dynamics are ordered and then time scaled by introducing parameter 'ϵ_i' to the left-hand side of the state equations. The slow variables are taken to be x_r and y_r, the horizontal components of relative velocity. Energy state (E) dynamics are faster than x_r and y_r dynamics but slower than altitude, heading, and flight path dynamics (z, ϕ and γ respectively). These later dynamics were not optimized in this study, and so are grouped as varying on the same (ϵ^2) time scale.

The cost function chosen for optimization should have elements of both minimum time and minimum fuel, in order to pick thrust magnitude so that the missile maneuvers efficiently from an energy point of view. For ramjet missiles, both thrust and rate of fuel flow are a complex function of velocity, fuel/air ratio, angle of attack, and altitude. The performance index chosen is of the form

$$J = \int (\sigma\, \theta\, w_f + (1 - \theta))\, dt \tag{7}$$

σ is a scaling parameter and $0 \le \theta \le 1$. $\theta = 0$ corresponds to minimum time and $\theta = 1$ corresponds to a minimum fuel cost function.

The following set of constraints are imposed on the optimal control solution:

$$L \le qSC_{Lmax} \quad \text{where} \quad L = \sqrt{L_1^2 + L_2^2} \tag{8}$$

$$L \le w\, G_{max} \tag{9}$$

$$T\,(v,\ (f/a)_{min},\ \alpha,\ h) \le T \le T\,(v,\ (f/a)_{max},\ \alpha,\ h) \tag{10}$$

$$v \ge v_{min} \tag{11}$$

C_{Lmax} is the maximum lift coefficient and is used to represent the stall angle of attack limit or a limit due to maximum control surface deflection. G_{max} represents a load factor limit which may arise due to a structural limit or a control surface actuator limit. T_{max} is imposed by the propulsion system and v_{min} is used to insure aerodynamic stability.

The optimal control problem is to drive the system from its present state to the intercept condition

$x_r\,(t_f) = 0$, $y_r\,(t_f) = 0$, $z = z_t$ (target altitude), t_f free while minimizing the performance index (7) subject to the equations (1) to (6) and constraints (8) to (11).

The optimality conditions are

$$\partial H/\partial L_1 = \partial H/\partial L_2 = \partial H/\partial (f/a) = 0 \tag{12}$$

where

$$\begin{aligned} H = {} & \lambda_x\, v \cos\gamma \cos\phi + \lambda_y\,(v \cos\gamma \sin\phi - v_t \cos\gamma_t) + \lambda_E\,(T - D)\,v/w \\ & + \lambda_z\, v \sin\gamma + \lambda_\phi\, g\, L_2/w\, v \cos\gamma + \lambda_\gamma\, g\,(L_1/w - \cos\gamma)/v + \theta\,\sigma\, w_f \\ & + 1 - \theta + \text{constraints} = 0 \end{aligned} \tag{13}$$

and where the co-state variables satisfy

$$\dot{\lambda}_x = -\,\partial H/\partial x = 0 \tag{14}$$

$$\dot{\lambda}_y = -\,\partial H/\partial y = 0 \tag{15}$$

$$\epsilon\,\dot{\lambda}_E = -\,\partial H/\partial E \tag{16}$$

$$\epsilon^2\dot{\lambda}_z = -\,\partial H/\partial z \tag{17}$$

$$\epsilon^2\dot{\lambda}_\phi = -\,\partial H/\partial \phi \tag{18}$$

$$\epsilon^2\dot{\lambda}_\gamma = -\,\partial H/\partial \gamma \tag{19}$$

The solution to this problem results in a two point boundary value problem. A simplified solution based on singular perturbation theory is developed.

Outer Solution

The outer solution (cruise solution) yields an optimum heading (ϕ^o), velocity (v^o), and altitude (z^o) based on the current target velocity. z is computed on the basis of a range matching condition and in conjunction with the energy rate boundary layer solution [6].

Taking the limit as $\epsilon \rightarrow 0$, the necessary conditions in Eqs. (1) to (6) and (12) to (19) are reduced to:

$$T = D, \ \gamma = 0, \ L_2 = 0, \ L_1 = w \tag{20}$$

$$\partial H_o / \partial E = \partial H_o / \partial z = \partial H_o / \partial \phi = 0 \tag{21}$$

$$\begin{aligned} H_o &= \lambda_x \, v \cos \phi + \lambda_y \, (v \sin \phi - v_t \cos \gamma_t) + \theta \, \sigma \, w_f + 1 - \theta \\ &\quad + \text{constraints} = 0 \end{aligned} \tag{22}$$

Note that the conditions in (12) are dropped since the constraints in (20) uniquely define L_1, L_2 and f/a at a given flight condition (z, v). The conditions in (21) imply that the fast variables (E, z and ϕ) take on the role of control variables. The constraint terms adjointed to the Hamiltonian always have a zero value, and only play a role in the variations in (21).

It can be shown [6] that the cruise velocity and altitude may be obtained from:

$$v^o, \ z^o = \underset{v,\ z}{\arg\max} \left\{ \frac{(v - v_t \cos \gamma_t \sin \phi)/\cos \phi}{\theta \, \sigma \, w_f + 1 - \theta} \right\} \begin{array}{l} T = D \\ L = W \\ \gamma = 0 \end{array} \tag{23}$$

Subject to the constraints in (8) to (11). Heading ϕ is a function of v and z and is established as the collision course heading in the horizontal plane. Using v^o from (23) the optimal heading is given by:

$$\phi_o^o = \sin^{-1} (v_t \cos \gamma_t \cos \lambda / v^o) + \lambda \tag{24}$$

where λ is the LOS angle. The corresponding co-states are found to be:

$$\lambda_x^o = - (\theta \, \sigma \, w_f^o + 1 - \theta) \cos \phi_o^o / (v^o - v_t \cos \gamma_t \sin \phi^o) \tag{25}$$

$$\lambda_y^o = \lambda_x^o \tan \phi^o \tag{26}$$

Energy Boundary Layer

In the energy boundary layer, an optimum altitude and thrust level are determined for climb and descent. In most cases the optimum thrust for climb and descent are close to maximum and minimum fuel flow, respectively. In general the climb and descent optimums are a function of E^o, z^o and the current missile energy (E).

The optimality conditions in the boundary layer are obtained by introducing the time transformation $\tau = t/\epsilon$ and again taking the limit $\epsilon \rightarrow 0$. Carrying this out on Eqs. (1) to (6) and (12) to (19), we obtain:

$$\gamma = 0, \ L_2 = 0, \ L_1 = w \tag{27}$$

$$\partial H_1 / \partial z = \partial H_1 / \partial \phi = \partial H_1 / \partial \, (f/a) = 0 \tag{28}$$

$$\begin{aligned} H_1 &= \lambda_x^o \, v \cos \phi + \lambda_y^o \, (v \sin \phi - v_t \cos \gamma_t) \\ &\quad + \lambda_E \, (T - D) \, v/w + \theta \, \sigma \, w_f + 1 - \theta + \text{constraints} = 0 \end{aligned} \tag{29}$$

It can be shown [6] that the optimal heading $\phi^1 = \phi^o$ and that the optimum altitude and fuel/air ratio schedule, z (E) and f/a (E), for climb and descent are obtained from:

$$z^1,\ (f/a)^1 = \underset{z,\ f/a}{\arg\max} \{k\,(T - D)\, v/w/[\lambda_x^o\, v \cos\phi^o + \lambda_y^o\,(v \sin\phi^o - v_t \cos\gamma_t) \tag{30}$$

$$+ \theta\, \sigma\, w_f + 1 - \theta]\}$$ where k = 1 for climb and k = -1 for descent.

Lift and Thrust Calculations

Since the remaining missile dynamics are not currently optimized, it is necessary to determine the lift components such that the missile follows ϕ^o and z^1 (climb and descent) as closely as possible, with a reasonable response transient. The horizontal component (L_2) is based on achieving a missile heading rate proportional to the heading error ($\delta\phi = \phi^o - \phi$), or

$$L_2 = (w\, v/g\, \tau_2)\, \delta\phi \tag{31}$$

where τ_2 is the time constant associated with the first order response transient. In the vertical plane we compute a desired flight-path angle (γ_D) such that altitude rate is proportional to the altitude error ($\delta z = z^1 - z$), or

$$\gamma_D = \delta z/\tau_1\, v \tag{32}$$

Then the vertical lift (L_1) is chosen such that the missile flight path angle rate is proportional to $\delta\gamma = \gamma_D - \gamma$, or

$$L_1 = (w\, v/g\tau_2)\, \delta\gamma + w \cos\gamma \tag{33}$$

The components of lift are then limited to satisfy G_{max} and angle-of-attack limits. Also, during ascent lift is limited to insure that the lift induced drag does not result in a negative value of energy rate.

The f/a control is computed based on the energy rate $\dot{E}^1$ from the first boundary layer solution. From (3), the desired value of thrust is

$$T = w\, \dot{E}^1/v + D \tag{34}$$

where D is determined for the lift components in Eqs. (31) and (33). f/a is computed by interpolating the propulsion for the current values of (v, T, α, h). Thrust is limited by the values corresponding to maximum and minimum fuel flow.

NUMERICAL RESULTS

The following test cases were run to check the effectiveness of the long range optimal control laws developed based on singular perturbation theory:

i) head-on
ii) tail chase
iii) beam
iv) variation in engine parameters

Conventional lofted trajectories ($\gamma(0) = 45°$) using a standard PN guidance law and maximum f/a were generated for cases i, ii and iii in order to weigh the merits of the optimal TMC control laws. These results are discussed in the sections to follow.

For each engagement, the launch weight of the missile is 596 lbs, the target initial velocity is 1200 ft/sec at an altitude of 25,000 ft, and the missile launch velocity is 970 ft/sec at 10,000 ft altitude. A rocket boost thrust of 8800 lbs is utilized for the first 3.1 seconds of flight to obtain ramjet ignition velocity.

Head-On Engagements

Altitude-time profiles for the head-on trajectories for nominal engine parameters are shown in Fig. 2. The launch range is 400,000 feet, and ramjet take-over occurs at $v = 2370$ ft/sec. The optimal trajectories for $\theta = 1.0$, 0.8 and 0.4, and the conventional lofted trajectory successfully engage the target. A comparison of the time of flight versus fuel remaining is also given in the figure. An example of the thrust history for each trajectory is given in Figure 3.

Tail Chase Engagements

The launch range for the tail chase engagements was modified to 300,000 ft., leaving unchanged the remaining launch geometry parameters. Trajectory profiles are displayed in Figure 4.

Beam Engagements

For the beam engagements (Figure 5) the launch range was reduced to 250,000 ft. The conventional trajectory achieved a high drag condition ($\alpha = \alpha$ max) and eventually reached dynamic pressure instability.

Engine Parameter Variations

These simulations investigated the effects on trajectory performance when perturbations in selected engine parameters were implemented. ± 10% variations were inserted about the nominal values of inlet capture area A_c and nozzle throat area A_5. Trajectories were generated for the 400,000 ft head-on engagement for $\theta = 1.0$. Altitude profiles are displayed in Figure 6.

It is evident that trajectory shaping is influenced by small variations in engine coefficients. In particular, it is apparent that both capture area and throat area have been optimized for this design.

CONCLUSIONS

An analysis tool has been developed that obtains ramjet missile system performance data for optimal implementation of aerodynamic and thrust magnitude control. The optimal control law is a nonlinear function of missile state, target state, and the missile aerodynamic and propulsion parameters. Singular perturbation analysis techniques were used to obtain a solution in this form. This effort both demonstrates the feasibility of on-line optimal control, and lays the groundwork for a new approach to ramjet missile design that permits simultaneous optimization of propulsion and guidance.

REFERENCES

1. Calise, A. J., "Extended Energy Management Methods for Flight Performance Optimization", AIAA 13th Aerospace Sciences Meeting, Pasadena, California, Jan. 20-22, 1975.

2. Calise, A. J., "Singular Perturbation Methods for Variational Problems in Aircraft Flight", IEEE Trans. on A.C. Vol. AC-21, No. 3, June 1976.

3. Calise, A. J., "Singular Perturbation Analysis for Optimal Thrust Control with Proportional Navigation Guidance", 1977 IEEE Conference on Decision and Control, New Orleans, Louisiana.

4. Aggarwal, R., Calise, A. J., Goldstein, F., "Singular Perturbation Analysis of Optimal Flight Profiles for Transport Aircraft", 1977 Joint Automatic Control Conference, San Francisco, California.

5. Goldstein, F., Sridhar, B., Calise, A. J., "Optimal Guidance and Thrust Magnitude Control for Air-Launched Tactical Missiles", Report R-229U, HQ USAF (AF/SAGF), Washington, D. C., 15 Sept. 1977.

6. Goldstein, F. and Calise, A., "Optimal Guidance and Thrust Magnitude Control for Air Launched Ramjet Powered Missiles". DRC Report No. R-278U (Vol. 1), Wright-Patterson AFB/RJT, Dayton, Ohio, April 1979.

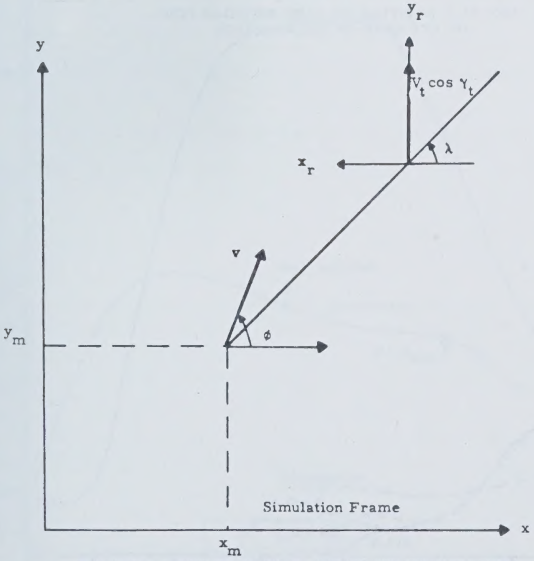
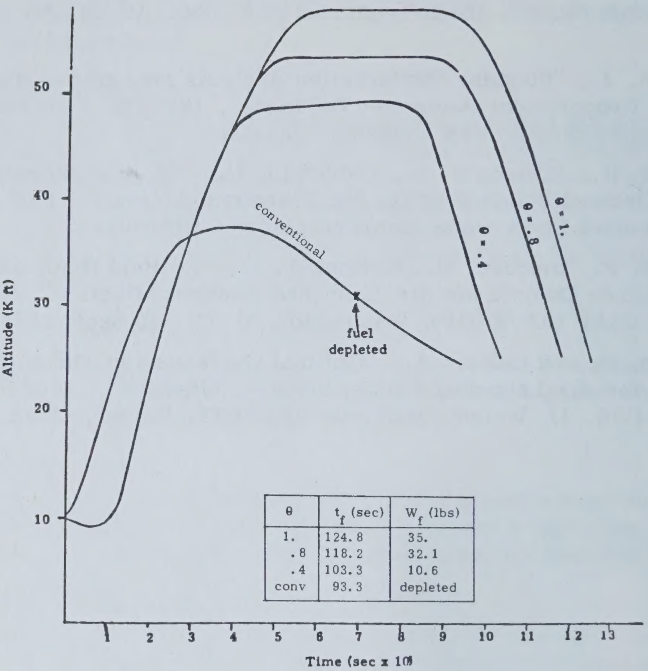


FIGURE 1 MISSILE-TARGET CONFIGURATION IN THE HORIZONTAL LONG RANGE ANALYSIS FRAME



θ	t_f (sec)	W_f (lbs)
1.	124.8	35.
.8	118.2	32.1
.4	103.3	10.6
conv	93.3	depleted

FIGURE 2 ALTITUDE VS. TIME PROFILES FOR
400 KFT HEAD-ON ENGAGEMENTS

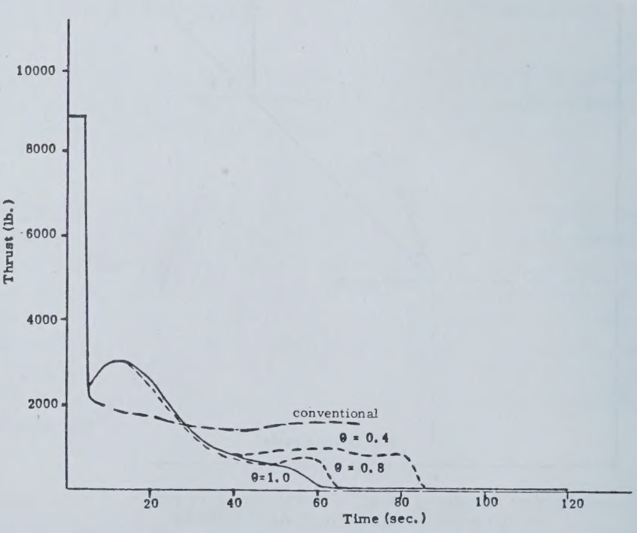


FIGURE 3 THRUST VS. TIME PROFILES FOR
400 KFT HEAD-ON ENGAGEMENTS

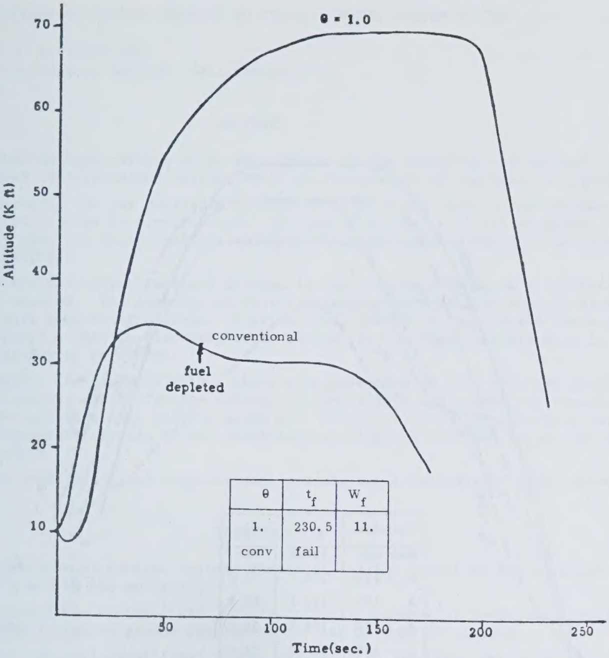


FIGURE 4 ALTITUDE VS. TIME PROFILES FOR 300 KFT TAIL CHASE ENGAGEMENTS

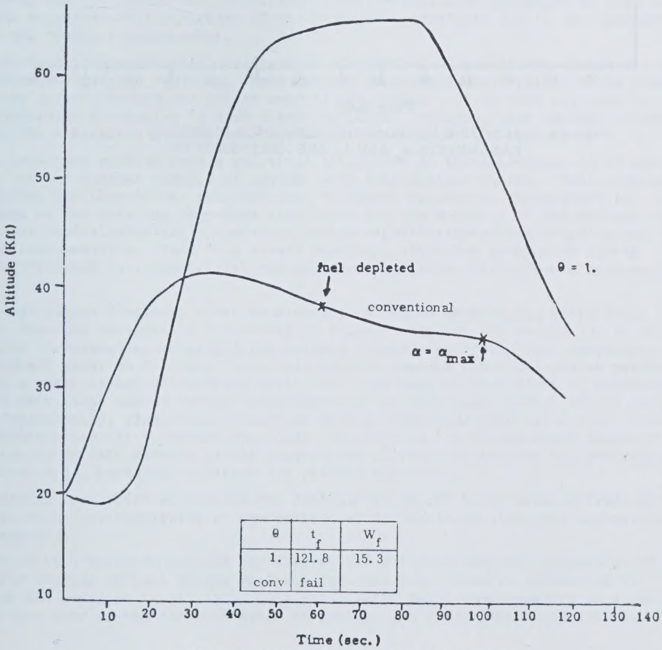
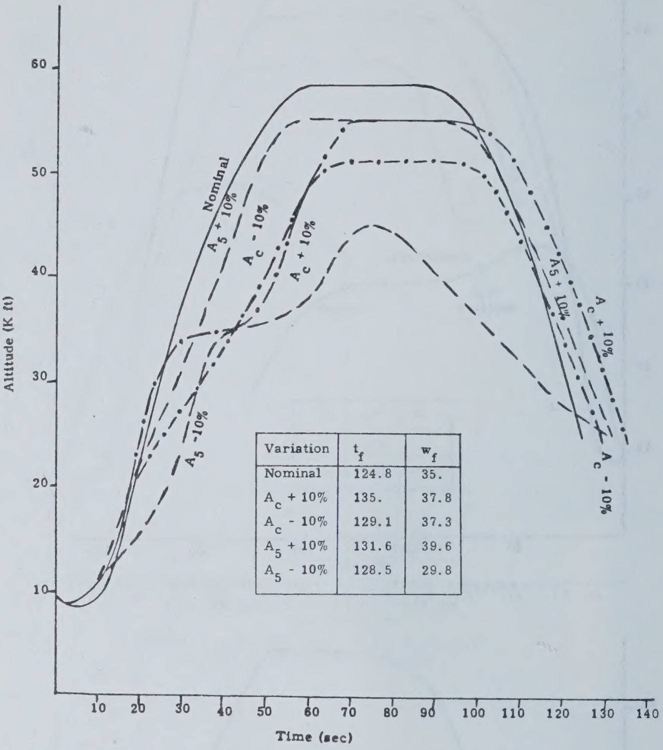


FIGURE 5 ALTITUDE VS. TIME PROFILES FOR 250 KFT BEAM ENGAGEMENTS



Variation	t_f	w_f
Nominal	124.8	35.
A_c + 10%	135.	37.8
A_c - 10%	129.1	37.3
A_5 + 10%	131.6	39.6
A_5 - 10%	128.5	29.8

Figure 6 ALTITUDE VS. TIME PROFILES WHEN ENGINE PARAMETERS A_c AND A_5 ARE PERTURBED 10%

APPLICATION OF OPTIMAL CONSTANT OUTPUT FEEDBACK TO FLIGHT CONTROL SYSTEM DESIGN

E. Y. SHAPIRO AND D. A. FREDRICKS
Lockheed-California Company, Burbank, California 91250,

ABSTRACT

This paper describes an applications oriented approach to the generation of optimal output feedback gains for linear time-invariant systems which is independent of the open loop stability.

The standard requirement for the provision of initial stabilizing output feedback gains for priming the computational process is circumvented. In lieu of initial stabilizing gains, the proposed algorithm employs the full state feedback solution which guarantees stability under the mild condition of stabilizability.

In this paper the optimal output feedback problem is cast in the setting of a constrained parameter optimization problem. The solution of this constrained optimization employs Hestenes' method of multipliers with some modifications. A primal-dual problem is considered where the primal minimization employs a Davidon-Fletcher-Powell method, and the dual maximization is accomplished via a quasi-Newton procedure.

This approach provides the designer with a means of suppressing to zero selected gains corresponding to accessible output, either for the purpose of simplifying the controller structure or because prior knowledge indicates that certain gains are "nonproductive". In addition, the designer can easily incorporate certain linear constraints, which the feedback gains will satisfy, into the proposed procedure.

Detailed algorithm description and computational results for a realistic flight control design problem are provided.

I. INTRODUCTION

In practice, the designer of control systems rarely enjoys the luxury of having the freedom to feed back the system's state in its entirely.

Lack of measurements of the entire state usually reflects economic or technical considerations which cause one to employ a limited sensor complement for the purpose of control.

In such a situation the well-established theory dealing with quadratic optimization under full state availability motivates the designer to employ observer-based compensator procedures in the deterministic setting or a Kalman filter in the stochastic setting. However, these state reconstructors, while allowing the use of the well-developed theory of quadratic optimization with full state feedback, tend to complicate the structure of the feedback controllers due to the introduction of dynamics into the feedback compensator.

This problem is further compounded when dealing with the control of non-linear systems, such as flight control systems for piloted vehicles, where several linearized small perturbation models are used to describe the system through its entire operational regime. It is then required to schedule the optimal controllers relative to each operating point. However, when dynamic compensators are introduced, the scheduling problem can become quite cumbersome and costly.

Consequently, an important problem from a practical standpoint is the determination of constant output feedback gains for the optimal control of systems with inaccessible states. This problem has been recently studied from the theoretical and numerical viewpoint by several researchers [1, 2, 3, 4, 5, 6]. However, most of the existing iterative algorithms for the solution of the optimal output feedback problem call for initializing the computation procedure with stabilizing output gains for systems which are open loop unstable. This is a severe handicap, since the problem of system stabilization by output feedback is a non-trivial one and could be quite costly from a computational viewpoint.

Furthermore, optimal output feedback, where each output is fed to each input, could still be structurally complex. Thus the designer is interested to further simplify the controller by eliminating some of the gains corresponding to accessible outputs, hence simplifying the controller. It is well known that feedback gains do not contribute uniformly to improve the total system performance. Moreover, only a small number of feedback gains carry the bulk of the effort of improving the performance, while many gains can be termed "nonproductive" as they have minimal effect on the system performance. Consequently, significant reduction of the controller complexity with corresponding minimal performance sacrifice becomes feasible. In addition, it is sometimes mandatory to force the feedback gains to satisfy certain linear constraints in order to achieve the desired performance characteristics (e.g., handling qualities for piloted aircraft).

It seems that a design tool which will allow the designer to weight the available trade-offs between performance vs. structure complexity of the controller is needed to allow for optimal/cost effective design procedures.

It is the purpose of this paper to outline the theoretical and computational procedures which resolve the problem of providing optimal output feedback for open loop unstable systems while simultaneously allowing the designer to eliminate some preselected gains corresponding to accessible outputs, and also to force some of the feedback gains to satisfy specified linear constraints.

III. PROBLEM FORMULATION

Consider the linear time invariant system

$$\dot{x} = Ax + Bu \qquad x(t=0) = x_0 \tag{1}$$

$$y = \tilde{C}x \tag{2}$$

where $x \in R_n$, $u \in R_m$, $y \in R_r$, and A, B, $\tilde{C}$ are real constant matrices of compatible order. Also let $\tilde{C}$ be of full rank, that is

$$\text{rank } \tilde{C} = r \tag{3}$$

With no loss of generality, we can assume that $\tilde{C}$ has the following partitioned form

$$\tilde{C} = [I_r \mid 0] \tag{4}$$

where I_r is the rxr identity matrix. If $\tilde{C}$ does not have the form of (4), it is always possible to find a similarity transformation which will yield $\tilde{C}$ of the form of (4) as long as (3) is satisfied.

For the system (1) and (2) subject to (3), find a control law of the form

$$u = -\bar{F}y = -\bar{F}\tilde{C}x \tag{5}$$

which minimizes the quadratic cost given by

$$J(\bar{F}) = E\left[\int_0^\infty (x'Qx+u'Ru)dt\right] \tag{6}$$

where $\bar{F}$ is an mxr constant matrix. Note that according to (4), y represents the first r components of the state x; that is

$$y = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_r \end{bmatrix} \tag{7}$$

For convenience, we introduce the matrix $\bar{C}$, where

$$\bar{C} = [0 \mid I_{(n-r)x(n-r)}] \tag{8}$$

and augment $\tilde{C}$ with $\bar{C}$ to form the matrix C, given by the partitioned form

$$C = \begin{bmatrix} \tilde{C} \\ \hline \bar{C} \end{bmatrix} \tag{9}$$

Note that C is the nxn identity matrix, and seemingly we are discussing a full state feedback problem.

The feedback matrix F corresponding to the matric C of (9) is an mxn matrix with special structure reflecting the threefold objective consisting of

a. Output feedback

b. Elimination of gains corresponding to accessible outputs

c. Forcing some gains to satisfy certain linear constraints

However, rather than forcing a certain structure on F, we may force certain conditions on the f_{ij}'s-the entries of the matrix F. To see that more clearly, let S(·) be a column stacking operator. When S(·) operates on a pxq matrix V, it yields a pq vector which is made of the columns of V. Proceed by defining the mn vector f according to

$$f = S(F) \tag{10}$$

and consider the three constant matrices W_1, W_2, W_3 with dimensions $q_1$xmn, $q_2$xmn and $q_3$xmn respectively. We can form the qxmn matrix W according to

$$W = \begin{bmatrix} W_1 \\ \hline W_2 \\ \hline W_3 \end{bmatrix} \tag{11}$$

where q is given by

$$q = q_1 + q_2 = q_3 \tag{12}$$

Our three objectives, which were outlined above, can be accommodated by incorporating the set of constraints

$$Wf = d \tag{13}$$

where d is a vector in R_q, into the optimization problem (6).

Note that if w_i' is the i-th row of W, then the i-th equation implied by (13) is

$$w_i' f = d_i \tag{14}$$

Thus, if w_i' is of the form

$$w'_i = (0\ 0\ 0\ .\ .\ .\ 0\ 1\ 0\ .\ .\ .\ 0) \tag{15}$$

and the corresponding $d_i = 0$, (14) implies that a certain f_{ij} should be set to zero. Consequently, (13) can be viewed as a structural constraint on F. To accomplish our objectives, the matrix W_1 will enforce an output feedback by having rows of the form of (15) with

$$d_i = 0 \qquad 1 \leq i \leq q_1 \tag{16}$$

The matrix W_2 will allow the designer to eliminate gains corresponding to accessible outputs by having rows of the form of (15) with the single non-zero entry of "1" in a position corresponding to the f_{ij} which is to be eliminated, with

$$d_i = 0 \qquad q_1 + 1 \leq i \leq q_1 + q_2 \tag{17}$$

The third component of W, namely W_3, will enable the designer to force some of the feedback gains to satisfy certain linear constraints; therefore the rows of W_3 and the respective d_i's have no special structure. Consequently, our optimization problem can now be stated as: Given the linear system of (1) with the output equation

$$y = C\,x \tag{18}$$

find the matrix F of

$$u = -F\,y \tag{19}$$

which minimizes (6) subject to the constraint

$$Wf = d \tag{20}$$

Before we proceed, it will be convenient to illustrate the formation of the matrix W for a specific example. Thus we consider the system (1), (2), and (4) with dimensions n = 4, m = 2 and r = 3. The feedback gain matrix F is of dimension 2x4. Note that the state x_4 is not accessible; further assume that it is desired to eliminate the feedback gain f_{12}, and it is also required to maintain the following linear relationship between f_{11} and f_{23}

$$f_{11} - 3\,f_{23} = 0 \tag{21}$$

The column stacking of the matrix F according to (10) yields the vector f, where

$$f = (f_{11}, f_{21}, f_{12}, f_{22}, f_{13}, f_{23}, f_{14}, f_{24})' \tag{22}$$

from which we readily determine the structure of W, that is

$$W = \begin{bmatrix} W_1 \\ \hdashline W_2 \\ \hdashline W_3 \end{bmatrix} = \left[\begin{array}{cccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hdashline 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hdashline 1 & 0 & 0 & 0 & 0 & -3 & 0 & 0 \end{array}\right] \begin{array}{l} \left.\vphantom{\begin{array}{c}0\\0\end{array}}\right\} W_1 \\ \left.\vphantom{0}\right\} W_2 \\ \left.\vphantom{0}\right\} W_3 \end{array} \tag{23}$$

and the vector d for this case is given by

$$d = 0 \tag{24}$$

Note that W_1 will cause the elimination of gains corresponding to inaccessible states, W_2 will cause the elimination of gains corresponding to accessible states, and W_3 takes care of the linear constraint (21).

III. PROBLEM DEVELOPMENT

The necessary conditions for the optimality of F have previously been derived [1, 5] by converting the dynamic optimization to a static optimization in the parameter space of the f_{ij}'s. It was shown that

$$J(F) = \mathrm{tr}P \tag{25}$$

where P satisfies the Lyapunov equation

$$(A-BFC)'P+P(A-BFC) = -(Q+C'F'RFC) \tag{26}$$

also, the gradient of J(F) with respect to F is given by [5]

$$\frac{\partial}{\partial F}\Big[J(F)\Big] = 2(RFCLC'-B'PLC) \tag{27}$$

where the matrix L satisfies the Lyapunov equation

$$(A-BFC)L+L(A-BFC)' = -I \tag{28}$$

Usually what one does is to employ (27) in a gradient search technique [6] and minimize J(F), provided an F_o which stabilizes $A-BF_oC$ can be found. However, in the present discussion, the matrix C is the identity matrix, and what is needed to initialize the gradient search is a full state feedback gain matrix, which stabilizes $A-BF_o$. To obtain such an F_o, one simply solves the Riccati equation corresponding to the quadruple A, B, Q, R

$$A'K+KA-KBR^{-1}B'K+Q = 0 \tag{29}$$

for the unique positive definite matrix K, and the respective F_o is given by

$$F_o = R^{-1}B'K \tag{30}$$

A solution for K can be readily obtained provided the pair $\{A, B\}$ is stabilizable, independent of the stability of the A matrix [7] .

The required solution of a Riccati equation does not represent additional computation effort, since the designer invariably will solve the full state feedback problem, thus establishing a data base for system performance. Consequently, the designer can always compare the performance of a system using output feedback to the performance of the ideal situation of full state feedback and exhibit the performance variation.

In view of the previous discussion, the problem addressed in this paper can be described by the following constrained minimization, that is:

$$\min_F J(F) = \min_F \mathrm{tr}\, P \tag{31}$$

subject to constraint

$$\xi = Wf - d = 0 \tag{32}$$

For convenience, we can replace (32) by an identical constraint given by

$$\|\xi\|^2 = \xi'\xi = 0 \tag{33}$$

The above minimization problem is equivalent to the following minimization problem [8]

$$\underset{F}{\text{minimize}} \left\{ J(F) + \mu \; \xi'\xi \right\} \tag{34}$$

subject to

$$\xi'\xi = 0 \tag{35}$$

for any $\mu > 0$, in particular, for reasons of convenience, let $\mu = 1/2 \; \gamma$ where $\gamma > 0$.

From a different point of view, one can adjoin the constraint (32) to the objective function (31) via a qx1 vector Lagrange multiplier λ, and obtain an unconstrained minimization problem, that is,

$$\underset{F}{\text{minimize}} \left\{ J(F) + \lambda'\xi \right\} \tag{36}$$

However, since minimization of J(F) is equivalent to the minimization of $\{ J(F) + 1/2 \lambda \xi'\xi \}$ subject to the same constraint, we can combine (34) with (36) and obtain a single unconstrained minimization problem of the form

$$\underset{F}{\text{minimize}} \left\{ J(F) + \lambda'\xi + 1/2 \, \gamma \, \xi'\xi \right\} \tag{37}$$

Note that minimization of (37) w.r.t. the matrix F simultaneously provides an optimal output feedback gain matrix having the prespecified structure as dictated by the matrix W.

IV. PROBLEM SOLUTION

The dual method is applied to the problem by defining the dual function

$$\Phi_\gamma(\lambda) = \min_{F} \left\{ J(F) + \lambda'\xi + 1/2 \, \gamma \, \xi'\xi \right\} \tag{38}$$

If F* is the matrix minimizing the right hand side of (38), then

$$\frac{\partial}{\partial \lambda} \left[\Phi_\gamma \right] = \xi^* = Wf^* - d \tag{39}$$

Furthermore, for sufficiently large γ, the Hessian of $-\Phi_\gamma$ approaches $\frac{1}{\gamma} I_{mr}$, where I_{mr} denotes the mr x mr identity matrix [8].

Therefore, [8] it is possible to develop a modified Newton's method for solving the dual problem, namely maximization of (38) w.r.t.λ. That is, using $\frac{1}{\gamma} I_{mr}$ as an approximate Hessian and ξ^* as a gradient, the following iterative scheme is applicable

$$\lambda(k+1) = \lambda(k) + \gamma \; \xi^* \tag{40}$$

Now, as noted in [9, p 527] , each $\Phi\gamma$ has the same local maximum λ^*. Therefore, it is possible to update the γ in some fashion, if this is advantageous. Again, [9] points out that the intermediate unconstrained minimization becomes increasingly ill-conditioned as γ increases, but the dual iteration has an increasingly faster convergence rate. Therefore, the key is some balanced updating of the γ. When the γ are updated in some way, the process becomes the method of multipliers.

The present algorithm employs the scheme suggested by Powell [10] and Buys [11] ; i.e., to multiply γ by a constant greater than "1" at the end of each unconstrained minimization for which the constraint violation, as measured by $\xi'\xi$, is not decreased by a certain factor.

The inner loop unconstrained minimization is solved as in [6] by a Davidon - Fletcher - Powell type algorithm with an appropriately modified gradient formulation, that is, if we let H(F) be defined as:

$$H(F) = J(F) + \lambda'\xi + 1/2 \, \gamma \xi'\xi = J(F) + \lambda'(Wf-d) + 1/2 \, \gamma (Wf-d)'(Wf-d) \tag{41}$$

Recalling that the mn vector f is obtained from the matrix F by stacking its columns by use of the stacking operator S(.), it is convenient, before the development of an expression for the gradient of H(F), to define the unstacking operator $S^{-1}{}_{m}(.)$. This operator maps mn column vectors into the space of mxn matrices by taking m components at a time and using them as columns of the mxn matrix. Thus, according to our previous definitions of f and F

$$S^{-1}{}_{m}(f) = F \tag{42}$$

Using the notion of the unstacking operator, it can be shown that the gradient of H(F) with respect to F is given by

$$\frac{\partial}{\partial F}\Big[H(F)\Big] = \frac{\partial}{\partial F}\Big[J(F)\Big] + \sum_{j=1}^{q} \lambda_j \, S^{-1}{}_{m}(w_j) + \gamma \, S^{-1}{}_{m}(W'Wf - W'd) \tag{43}$$

where λ_j is the j-th component of the Lagrange multiplier λ, and w_j' is the j-th row of the matrix W.

The gradient of J(F) with respect to F in (43) is given by (27), but since in our case the matrix C is the identity matrix, we obtain

$$\frac{\partial}{\partial F}\Big[J(F)\Big] = 2\ (RFL-B'PL) \tag{44}$$

V. ALGORITHM DESCRIPTION

Based on the above discussion, an iterative algorithm for the determination of optimal output feedback for stable or unstable plants will now be briefly described. In what follows, recall that the matrix C is the n-th order identity matrix, and that output feedback, with the possibility of elimination of gains corresponding to accessible outputs and/or forcing some feedback gains to satisfy some linear constraints, is obtained by the special structure of the matrix W.

Step 1: Input the matrices: A(nxn), B(nxm), Q(nxn), R(mxm), W(qxmn). The qxl vector d, and the scalars β, ϵ, M.

Step 2. Initialize counter i = 0. Set the initial Lagrange Multiplier Vector $\lambda(0)$ to the qxl null vector, and select $\gamma(0)$ to be some "large" positive scalar.

Step 3. Use a Riccati solver to solve (29) and (30) to obtain the initial mxn feedback gain matrix F(0).

Step 4: Form

$$\xi(i) = Wf(i) - d \tag{45}$$

Step 5: If $\quad \xi'(i)\,\xi(i) < \epsilon \quad$ STOP
Else go on.

Step 6: Perform the inner loop minimization with respect to F, using the Davidon - Fletcher-Powell algorithm described in [6]. The cost function, recalling that $C = I_n$, is given by

$$H\big[F(i)\big] = J\big[F(i)\big] + \lambda'(i)\,\xi(i) + 1/2\,\gamma\,\xi'(i)\,\xi(i) \tag{46}$$

where

$$J\ \big[F(i)\big] = \mathrm{tr}\ P(i) \tag{47}$$

and P(i) is a solution to

$$\big[A-BF(i)\big]'\ P(i) + P(i)\big[A-BF(i)\big] = -\big[Q+F'(i)RF(i)\big] \tag{48}$$

and the gradient of H [F(i)] with respect to F is given by

$$\frac{\partial}{\partial F}\Big\{H\big[F(i)\big]\Big\} = \frac{\partial}{\partial F}\ J\Big\{F(i)\Big\} + \sum_{j=1}^{q} \lambda_j(i)\ S_m^{-1}\big[w_j\big] + \gamma(i)\ S_m^{-1}\big[W'\,\xi(i)\big] \tag{49}$$

and

$$\frac{\partial}{\partial F}\Big\{J\ \big[F(i)\big]\Big\} = 2\big[R\ F(i)\ \ L(i) - B'P(i)\ L(i)\big] \tag{50}$$

where L(i) is a solution of

$$\big[A-BF(i)\big]\ L(i) + L(i)\big[A-BF(i)\big]' = -I \tag{51}$$

Increment the counter for i by setting i ⟶ i+1, and denote the solution to the inner loop minimization by F(i).

Step 7: Update the Lagrange multiplier vector according to

$$\lambda(i+1) = \lambda(i) + \gamma(i)\,\xi(i) \tag{52}$$

Step 8. If

$$\frac{\|\xi(i-1)\|^2}{\|\xi(i)\|^2} > M$$

Then

$$\gamma(i) = \gamma(i-1)$$

Else

$$\gamma(i) = \beta\,\gamma(i-1)$$

Step 9. Go to Step 4.

VI. NUMERICAL EXAMPLE

We illustrate the method with a simplified flight control problem; specifically, we consider the inner loop lateral axis design problem taken from [12]. The dynamics are

$$\dot{x} = Ax + Bu \tag{53}$$

$$y = \tilde{C}\,x \tag{54}$$

with

$$x = \begin{array}{cl} \left[\begin{array}{c} p_s \\ r_s \\ \beta \\ \delta_a \\ \delta_r \\ \Phi \end{array}\right] & \begin{array}{l} \text{stability axis rool rate} \\ \text{stability axis yaw rate} \\ \text{angle of sideslip} \\ \text{aileron deflection} \\ \text{rudder deflection} \\ \text{bank angle} \end{array} \end{array}$$

$$u = \begin{array}{cl} \left[\begin{array}{c} \delta_{r_c} \\ \delta_{a_c} \end{array}\right] & \begin{array}{l} \text{rudder command} \\ \text{aileron command} \end{array} \end{array}$$

and

$$A = \begin{bmatrix} -.746 & .387 & -12.9 & 6.05 & .952 & 0 \\ .024 & -.174 & .4 & -.416 & -1.76 & 0 \\ .006 & -.999 & -.058 & -.0012 & .0092 & .0369 \\ 0 & 0 & 0 & -5.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -10.0 & 0 \\ 1.0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{55}$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 10.0 \\ 20.0 & 0 \\ 0 & 0 \end{bmatrix} \tag{56}$$

Here we have destabilized the matrix A from [12] to exhibit the generation of optimal output feedback gains of open loop unstable systems. The change of A corresponds to an aft motion of the center of gravity of the aircraft. The open loop eigenvalues corresponding to A are given by

$$\left.\begin{array}{rcl} \lambda_1 & = & -.200 \text{ (Spiral Mode)} \\ \lambda_2 & = & -.778 \text{ (Roll Subsidence Mode)} \\ \lambda_{3,\,4} & = & .00013 \pm j.704 \text{ (Dutch Roll Mode)} \\ \lambda_5 & = & -5.0 \text{ (Actuator)} \\ \lambda_6 & = & -10.0 \text{ (Actuator)} \end{array}\right\}$$

The actuator bandwidths have been chosen deliberately low to illustrate the selection of an appropirate bandwidth by means of feedback.

Since we consider the inner loop design problem, the bank angle is an inaccessible state. Consequently,

$$\tilde{C} = (I_5 \mid 0) \tag{57}$$

The solution of the full state feedback problem, given the weighting matrices from [12]

$$Q = H'_o H_o \tag{58}$$

where

$$H_o = \begin{bmatrix} -.131 & -.612 & 1.64 & 0 & 1.0 & .0175 \\ .567 & .160 & -2.39 & 1.0 & 0 & .0303 \end{bmatrix} \tag{59}$$

and

$$R = \begin{bmatrix} .01 & 0 \\ 0 & .01 \end{bmatrix} \tag{60}$$

is given by

$$F = \begin{bmatrix} -1.28 & -6.26 & 16.47 & -.0044 & 9.56 & .179 \\ 5.64 & 1.83 & -24.50 & 9.84 & -.0022 & .298 \end{bmatrix} \tag{61}$$

The corresponding closed loop eigenvalues are

$$\left.\begin{aligned} \lambda_1 &= -.0504 \\ \lambda_2 &= -4.00 \\ \lambda_{3,4} &= -.635 \pm j\,1.41 \\ \lambda_5 &= -100 \\ \lambda_6 &= -200 \end{aligned}\right\} \tag{62}$$

For comparison, using the same weighting matrices, the algorithm described in Section V was exercised to produce the output feedback solution. Recall that the bank angle is an inaccessible state. We simplify the controller structure by eliminating the actuator coupling gains f_{14} and f_{25} . Feedback of roll rate to rudder and yaw rate to aileron is known to be "unproductive". Hence, elimination of f_{11} and f_{22} further simplifies the controller structure. Finally, we impose the constraint

$$f_{23} = -2\, f_{13} \tag{63}$$

The solution for the output feedback gains was found to be:

$$F = \begin{bmatrix} 0 & -5.25 & 12.9 & 0 & 6.70 \\ 7.33 & 0 & -25.8 & 13.4 & 0 \end{bmatrix} \tag{64}$$

and the corresponding eigenvalues are

$$\left.\begin{aligned} \lambda_1 &= -.0065 \\ \lambda_2 &= -4.06 \\ \lambda_{3,4} &= -.749 \pm j\,1.46 \\ \lambda_5 &= -135.8 \\ \lambda_6 &= -142.7 \end{aligned}\right\}$$

Note that these roots are in good agreement with the eigenvalues of the full state feedback solution while the controller structure has been simplified by 50 percent.

VII. CONCLUSIONS

An approach for generating optimal constant output feedback gains which is independent of open loop stability has been presented. In addition, this approach allows the designer to further simplify the controller structure by eliminating gains corresponding to accessible outputs, yet maintaining optimality of the remaining ones. Furthermore, the designer is able to incorporate certain linear constraints on the feedback gains to accommodate desired performance characteristics. Computational procedures and a numerical example have been presented.

REFERENCES

[1] W. S. Levine and M. Athans, "Determination of the Optimal Constant Feedback Gains for Linear Multivariable Systems," IEEE Transactions on Automatic Control, Volume AC-15, No. 1, February, 1970, pp. 44-48.

[2] S. S. Choi and H. R. Sirisena, "Computation of Optimal Output Feedback Gains for Linear Multivariable Systems," IEEE Transactions on Automatic Control, Vol. AC-19, No. 3, June, 1974, pp. 257-258.

[3] H. P. Horisberger and P. R. Belanger, "Solution of the Optimal Constand Output Feedback Problem by Conjugate Gradients, "IEEE Transactions on Automatic Control, Vol. AC-19, August, 1974, pp. 434-435.

[4] C. H. Knapp and S. Basuthakur, "On Optimal Output Feedback, "IEEE Transactions on Automatic Control, Vol. AC-17, No. 6, December 1972, pp. 823-825.

[5] J. Mendel, "A Concise Derivation of Optimal Constant Limited State Feedback Gains, "IEEE Transactions on Automatic Control, Vol. AC-19, No. 4, August, 1974, pp. 447-448.

[6] E. Y. Shapiro and D. A. Fredricks, "An Algorithm for the Generation of Optimal Output Feedback Gains for Linear Time Invariant Systems," Proceedings of the Conference on Information Sciences and Systems, The Johns Hopkins University, Baltimore, Maryland, March 29-31, 1978, pp. 467-472.

[7] H. Kwakernaak and R. Sivan, Linear Optimal Control Systems, Wiley-Interscience, 1972.

[8] D. G. Luenberger, Introduction to Linear and Nonlinear Programming, Addison-Wesley, 1973.

[9] D. P. Bertsekas, "Combined Primal-Dual and Penalty Method for Constrained Minimization," Siam Journal on Control, Vol. 13, May, 1975, pp. 521-542.

[10] M. J. D. Powell, "A Method for Nonlinear Constraints in Minimization Problems, "Optimization, R. Fletcher, ed., Academic Press, New York, 1969, pp 283-298.

[11] J. D. Buys, "Dual Algorithms for Constrained Optimization, "Phd. Thesis, Rijksuniversiteit de Leiden, the Netherlands, 1972.

[12] C. A. Harvey and G. Stein, "Quadratic Weights for Asymptotic Regulator Properties, "IEEE Transactions on Automatic Control, Volume AC-23, No. 3, June, 1979, PP. 378-387.

OPTIMAL SENSITIVITY AND FEEDBACK: WEIGHTED SEMINORMS, APPROXIMATE INVERSES, AND PLANT INVARIANT SCHEMES

GEORGE ZAMES
Dept. of Electrical Eng.
McGill University
3480 University Street
Montreal, Quebec H3A 2A7
Canada

1. INTRODUCTION AND ABSTRACT

(This brief is a summary of a report which can be obtained from the author. Proof are in the report.) The objectives of this paper are to formulate the problem of sensitivity reduction by feedback as an optimization problem; to derive some properties of sensitivity reducing schemes; and to show that plant uncertainty decreases the ability of feedback to reduce sensitivity. In the general theory, systems will be represented by linear, input-output mappings belonging to a normed algebra. The theory will be illustrated by simple examples of plants whose frequency responses have a r.h.p. (right-half-plane) zero, and multivariable systems without such zeros. Optimal sensitivity will be shown to be constrained by r.h.p. zeros, in whose absence sensitivity can be made arbitrarily small.

Our approach will be based on the use of an auxilliary 'weighted' norm to measure sensitivity, and on the concept of an approximate inverse. We shall make extensive use of an equivalence between feedback and model reference schemes, based on the well known fractional transformation. The transformation replaces unstable feedbacks by stable operators. We shall also employ the notion of a plant-invariant scheme: For plants in a sphere of uncertainty centered around a nominal value, an invariant scheme leaves the plant unchanged whenever it assumes the nominal value, but shrinks the sphere of uncertainty otherwise.

Motivation

A few observations might serve to motivate this work. The methods of 'classical' control theory, which rely on lead-lag compensators to reduce sensitivity, have continued to dominate many areas of design, despite the unquestioned success of the state-space and W.H.K. (Wiener-Hopf-Kalman) approaches. It is becoming clear that the W.H.K. and classical theories in many cases should not be viewed as alternative solutions to a common problem, but as methods for two disparate categories of problems. The W.H.K. approach is essentially a projection method for finding the best quadratic approximation to a single, known vector or to a single random process of precisely known covariance properties. In many problems, however, the vector to be approximated is an uncertain element of a prescribed set, or a random process of uncertain covariance belonging to a prescribed set of covariances. Such problems, as well as problems of nonparametric plant uncertainty, appear to be untractable by W.H.K. techniques. Instead, they are now solved by empyrical, feedback methods resembling those of sensitivity theory.

Sensitivity theory is also one of the few tools available for the study of organization structure (feedback vs. open-loop, aggregated vs. disaggregated, etc.) For example, feedback can reduce the complexity of identification [6] roughly for the same reasons that it can reduce sensitivity.

Despite its usefulness, classical sensitivity/compensation has been difficult to pin down in a mathematical theory. It has been elusive partly because the purpose of compensation usually has not been presented as a well defined mathematical objective amenable to optimization. Without a clear notion of optimality, it is also hard to draw conclusions about the effects of organization on sensitivity. For example, the observation that aggregation reduces performance for a particular control law conveys little about aggregation, unless the control law is optimal.

Weighted Seminorms and Approximate Inverses

One way of defining the optimal sensitivity of a feedback system, and of addressing some of the issues mentioned in Sect. 1A, is in terms of an induced norm of the sensitivity operator. However, it will be shown in Sect. 3 that the primary norm of an operator in a normed algebra is useless for this purpose. Perhaps that is why operator norm optimization has not been pursued extensively in the past. Instead, we shall introduce a weighted seminorm alongside the primary norm, and use the former to define the concept of an approximate inverse. Sensitivity reduction will be possible iff there is such an inverse.

2. SPACES AND ALGEBRAS

The algebra H_e^∞ (of *causal*, not necessarily stable, complex frequency responses,) consists of functions $\hat{p}:\mathbb{C}\to\mathbb{C}$, $\hat{p}(s)$ analytic and bounded, i.e., $|\hat{p}(s)| \leqslant \text{Const.}$, for $s = \sigma+j\omega$ in some closed half-plane $\text{Re}(s) \leqslant \sigma_p$ which may depend on $\hat{p}$. The Banach algebra H^∞ consists of functions $\hat{p}$ in H_e^∞ for which $\sigma_p \leqslant 0$, under the **norm** $\|\hat{p}\| \triangleq \sup\{|\hat{p}(s)| : \text{Re}(s) \leqslant 0\}$. The algebras H_{eo}^∞ and H_o^∞ (of *strictly causal* frequency responses) consists of functions $\hat{p}$ of H_e^∞ and H respectively, for which $\hat{p}(s)\to 0$ as $|s|\to\infty$ in $\text{Re}(s) \geqslant \sigma_p$.

The linear space H_e^2 consists of functions $\hat{u}:\mathbb{C}\to\mathbb{C}$, analytic in some open half plane $\text{Re}(s)>\sigma_u$ in which they are uniformly L_2-bounded, i.e., $\int_{-\infty}^{\infty}|\hat{u}(\sigma+j\omega)|^2 d\omega \leqslant \text{Const.}$ for each $\sigma > \sigma_u$, and defined on the boundary by the limit, $\hat{u}(\sigma_u+j\omega) = \lim_{\sigma\to\sigma_u} \hat{u}(\sigma+j\omega)$, where $\sigma\to\sigma_u$ from the right. The Banach space H^2 consists of functions $\hat{u}$ in H_e^2 for which $\sigma_u \leqslant 0$ under the norm $\|\hat{u}\| = \sup\{(\int_{-\infty}^{\infty}|u(\sigma+j\omega)|^2 d\omega)^{\frac{1}{2}} : 6>0\}$.

For each algebra of frequency responses defined above, we define a corresponding and homomorphic algebra of operators denoted by a bordered capital, acting in H^2 in the stable case, or H_e^2 in the general case. For example, $\mathbb{H}^\infty$ is the algebra of *stable causal operators* $P:H^2 \to H^2$, $\widehat{Pu}(s) = \hat{p}(s)\hat{u}(s)$, where $\hat{p}$ is in H^∞ and s in $\text{Re}(s) \geqslant 0$. The norm on $\mathbb{H}^\infty$ is defined by $\|P\|_{\mathbb{H}^\infty} = \|\hat{p}\|_{H^\infty}$.

$\mathcal{X}$ is a linear space whose elements will be called *inputs* or *outputs*. $\mathbb{A}$ is a linear algebra of linear mappings $p:\mathcal{X}\to\mathcal{X}$ with identity I, whose elements will be called *causal operators*. $\mathbb{A}_s$ is a radical of $\mathbb{A}$, i.e., a proper nontrivial subalgebra of $\mathbb{A}$ with the property that for any P in $\mathbb{A}_s$ and F in $\mathbb{A}$ the products PF and FP are in $\mathbb{A}_s$, and the inverse $(I+P)^{-1}$ exists in $\mathbb{A}$. The elements of $\mathbb{A}_s$ will be called strictly causal *operators*. The assumed properties of a radical ensure that solutions to feedback equations involving P in $\mathbb{A}_s$ exist and are unique. An example of $\mathcal{X}$, $\mathbb{A}$, and $\mathbb{A}_s$ is provided by the input space H_e^2, the algebra of frequency response operators $\mathbb{H}_e^\infty$, and the subalgebra $\mathbb{H}_{eo}^\infty$.

$\mathcal{B}$ is a Banach subspace of $\mathcal{X}$ whose elements will be called *bounded* inputs or outputs (for example, H^2 or $L_2(0,\infty)$). $\mathbb{B}$ is a subalgebra of $\mathbb{A}$ containing the identity I, whose elements will be called *stable* causal operators, under the following assumption: The norm of any P in $\mathbb{B}$ is the $\mathcal{B}$-induced norm, $\|P\| = \sup\{\|Pu\|/\|u\|:\ u \text{ in } \mathcal{B},\ u\neq 0\}$, the sup being finite . If $\mathbb{H}^{\infty}_{e}$ is taken as an example of $\mathbb{A}$, then $\mathbb{H}^{\infty}$ is an example of $\mathbb{B}$. $\mathbb{B}_s$ is the subalgebra of $\mathbb{B}$ obtained by intersecting $\mathbb{A}_s$ and $\mathbb{B}$, consisting of *strictly-causal stable* operators. It will be assumed that $\mathbb{B}_s$ has the *small-gain property*, i.e., for any P in $\mathbb{B}_s$, if $\|P\|<1$ then $(I+P)^{-1}$ is in $\mathbb{B}$.

3. THE MODEL REFERENCE TRANSFORMATION

Consider the system of Fig. 2. The plant input v, output y, and disturbance d are all in $\mathcal{X}$ and satisfy the equation

$$y = Pv + d \qquad (3.1a) \qquad\qquad v = -Fy \qquad (3.1b)$$

in which P and F are operators in $\mathbb{A}_s$. We shall refer to (3.1) as a *feedback scheme* with plant P and feedback F. Since P is strictly causal, the inverse $(I+PF)^{-1}$ exists in $\mathbb{A}$. Therefore, for each d in $\mathcal{X}$, Eqs. (3.1) have unique solutions for v and y in $\mathcal{X}$. Let $K_{32}: \mathcal{X}\to\mathcal{X}$ denote the "closed loop" operator mapping d to v. The set of c.l. operators for (3.1) consists of: $K_{22} = (I+FP)^{-1}$, $K_{23} \triangleq PK_{32}$, $K_{33} = (I+PF)^{-1}$ and K_{32}. The feedback scheme (3.1) will be called *c.l. stable* if K_{ij} is in $\mathbb{B}$ for i,j=2 or 3.

The flow graph of Fig. 3 is described by the equations

$$y = Pv + d \qquad (3.3a) \qquad\qquad v = -Q(y-P_1v) \qquad (3.3b)$$

in which y, v and d are in $\mathcal{X}$, and P, P_1, Q are in $\mathbb{A}_s$. (3.3) will be called a *model reference scheme* with *comparator* Q.

The two sets of equations (3.1) and (3.3) are called *equivalent* iff every input-output triple (d,v,y) in $\mathcal{X}^3$ satisfying (3.1) satisfies (3.3), and vice versa. Their equivalence will be established under the assumption that the equations

$$Q = F(I+P_1F)^{-1} \qquad (3.4a) \qquad\qquad F = Q(I-P_1Q)^{-1} \qquad (3.4b)$$

hold. *Assumption*. *For the present, assume that $P=P_1$; i.e., there is no plant uncertainty*.

Theorem 1 (a) Any feedback scheme (3.1) with feedback F in $\mathbb{A}_s$ is equivalent to a model reference scheme (3.3) with comparator Q in $\mathbb{A}_s$, and vice versa, where F and Q are related by Eqs. (3.4). If (3.4) hold, then:
(b) If d and y satisfy either the feedback or model reference equations they satisfy the equation

$$y = (I-P_1Q)d \ ; \qquad (3.6)$$

(c) whenever the feedback scheme is closed-loop stable, Q of the model reference scheme is stable (though the feedback F may be unstable).
Definition: The operator $(I-P_1Q)$ appearing in (3.6) will be called the *sensitivity operator* and denoted by E.

4. WEIGHTED SEMINORMS AND APPROXIMATE INVERSES

Let $\|\cdot\|_\ell$ be a seminorm defined on a $\|\cdot\|$-normed linear space $\mathcal{Y}$, and $\|\cdot\|_r$ a seminorm on a nontrivial subspace $\mathcal{Y}_r$ of $\mathcal{Y}$. The seminorm $\|\cdot\|_r$ is said to *dominate* $\|\cdot\|_\ell$, denoted by $\|\cdot\|_\ell \leqslant \|\cdot\|_r$, iff $\|y\|_\ell \leqslant \|y\|_r$ for all y in $\mathcal{Y}_r$. $\|\cdot\|_\ell$ and $\|\cdot\|_r$ will be called *left* and *right* seminorms iff they satisfy the respective conditions $\|\cdot\|_\ell \leqslant \overline{\|\cdot\|}$ and $\overline{\|\cdot\|} \leqslant \|\cdot\|_r$

A *weighted seminorm* $\|\cdot\|_W$ *on the space* $\mathbb{B}$ is any seminorm with the property that $\|\cdot\|_W \leqslant \overline{\|\cdot\|}$. A weighted seminorm $\|\cdot\|_W$ on $\mathbb{B}$ is *induced* by a pair $(\|\cdot\|_\ell, \|\cdot\|_r)$ of left and right seminorms on the space $\mathcal{B}$ (of outputs) and subspaces $\mathcal{B}_r$ of $\mathcal{B}$ (of inputs) respectively, iff $\|\cdot\|_W$ is defined by the equation $\|A\|_W = \sup\{\|Au\|_\ell / \|u\|_r : u \text{ in } \mathcal{B}_r \text{ and } \|u\|_r \neq 0\}$ for any A in $\mathbb{B}$. For example, let W_ℓ and W_r be linear mappings in $\mathcal{X} \times \mathcal{X}$, each of unit $\mathcal{B}$-induced norm. A left seminorm is defined on $\mathcal{B}$ by the equation $\|y\|_\ell \triangleq \|W_\ell y\|$, and is a true norm whenever Wy = 0 implies y = 0. Let $\mathcal{B}_r$ be the range of W_r; if W is 1:1, a right seminorm is defined on $\mathcal{B}_r$ by the equation $\|u\|_r \triangleq \|W_r^{-1}u\|$. The pair $(\|\cdot\|_r, \|\cdot\|_\ell)$ induces the weighted seminorm $\|A\|_W = \|W_\ell A W_r\|$ on the space $\mathbb{B}$.

If $\mathbb{B}_r$ is any subspace of $\mathbb{B}$, let $\mathbb{B}.\mathbb{B}_r$ denote the space of products {CD: C in $\mathbb{B}$ and D in $\mathbb{B}_r$}. A *weighted seminorm* on the space of products $\mathbb{B}\cdot\mathbb{B}_r$ is a weighted seminorm on the space $\mathbb{B}\cdot\mathbb{B}_r$ with the following additional property: There is a left seminorm $\|\cdot\|_{W\ell}$ defined on $\mathbb{B}$, a seminorm $\|\cdot\|_{W1}$ defined on $\mathbb{B}_r$, and the inequality $\|CD\|_W \leqslant \|C\|_{W\ell}\,\|D\|_{W1}$ holds for all C in $\mathbb{B}$ and D in $\mathbb{B}_r$.

Finally, a *symmetric* weighted seminorm on the *algebra* $\mathbb{B}$ is a weighted seminorm on the space $\mathbb{B}$ with the added property that both inequalities $\|CD\|_W \leqslant \|C\|_W\|D\|$ and $\|CD\|_W \leqslant \|C\|\cdot\|D\|_W$ hold for all C and D in $\mathbb{B}$.

Let $\|\cdot\|_W$ be a fixed, weighted seminorm on the space $\mathbb{B}$. For any operator P in $\mathbb{A}$, an *approximate right inverse* (in $\mathbb{B}_s$) of P is any operator Q in $\mathbb{B}_s$ for which $\|I-PQ\|_W < \|I\|_W$; the *right singularity measure* (in $\mathbb{B}_s$) of P (under $\|\cdot\|_W$), denoted by $\mu(P)$, is $\mu(P) = \inf\{\|I-PQ\|_W: Q \text{ in } \mathbb{B}_s\}$. In general, $0 \leqslant \mu(P) \leqslant \|I\|_W$.

Optimal Sequences: Symmetric Case

Let $\|\cdot\|_W$ be a fixed, symmetric weighted seminorm on the algebra $\mathbb{B}$. A *weighted identity sequence* (widseq) is any sequence I_m, m=1,2,... in $\mathbb{B}$ with the property that $\|I_m\|=1$ and for A in B, $\|A-I_nA\|_W$ and $\|A-AI_n\|_W$ approach 0 as $n\to\infty$. For any P in $\mathbb{A}$, a sequence Q_n, n=1,2,..., is a *right weighted inverting sequence* (winvseq) iff PQ_n is a widseq.

The following Lemma allows some optimization problems to be decomposed into simpler ones. Let $\|\cdot\|_a \geqslant \|\cdot\|_b$ be a pair of symmetric, weighted seminorms on the algebra $\mathbb{B}$. Let P_a in $\mathbb{B}_s$ and P_b in $\mathbb{B}$ have singularity measures $\mu_a(P_a)$ and $\mu_b(P_b)$ under $\|\cdot\|_a$ and $\|\cdot\|_b$ respectively and optimal sequences $\{Q_{am}\}_{m=1}^{\infty}$ in $\mathbb{B}_s$, $\{Q_{bn}\}_{n=1}^{\infty}$ in $\mathbb{B}$, i.e., $\|I-P_iQ_{in}\|_i \to \mu_i(P_i)$ as $n\to\infty$ for i=a or b.

Lemma 5.1 If $\mu_a(p_a)=0$, then $\mu_b(P_bP_a)=\mu_b(P_b)$, and P_bP_a has an optimal sequence of $\|\cdot\|_b$-weighted right approximate inverses in $\mathbb{B}_s$ of the form $Q_{an_m}Q_{bm}$, where for any m=1,2,..., n_m is a sufficiently large integer.

5. SENSITIVITY TO DISTURBANCES AND APPROXIMATE INVERTIBILITY

Consider the feedback scheme, Eqs. 3.1, and suppose the plant P in $\mathbb{B}$ equals the nominal P_1. Let $\|\cdot\|_r$ be a fixed right seminorm defined on some subspace $\mathcal{B}_r$ of (inputs), $\|\cdot\|_\ell$ a fixed left seminorm on (outputs) , and $\|\cdot\|_W$ the resulting weighted seminorm induced on $\mathbb{B}$. The *sensitivity to disturbances* η_1 is defined by the equation $\eta_1 \triangleq \sup\{\|y\|_\ell / \|d\|_r : d \text{ in } \mathbb{B}_r, \|d\|_r \neq 0\}$. From (3.2a) it follows that $\eta_1 = \|(I+P_1F)^{-1}\|_W$. Whenever the equivalence equations (3.4) hold, η_1 also equals $\|I-P_1Q\|_W$.

Theorem 2 (a) A necessary and sufficient condition for the existence of a feedback F in $\mathbb{A}_s$ for which (3.1) is closed-loop stable and the sensitivity η_1 is less than the open-loop value η_o, is that P_1 have an approximate (stable) right inverse Q in $\mathbb{B}_s$.

(b) For any $\varepsilon>0$, there is a feedback F in $\mathbb{A}_s$ for which (3.1) is closed-loop stable and for which $\eta_1<\mu(P_1)+\varepsilon$, but none for which (3.1) is closed-loop stable and $\eta_1<\mu(P_1)$, where $\mu(P_1)$ is the measure of right singularity in $\mathbb{B}_s$ of P_1 under $\|\cdot\|_W$.

An Example of Sensitivity Optimization

Let P_1 in $\mathbb{H}_o^\infty$ be any plant with a single Re(s) > 0 zero at $s=\beta$, subject to the high frequency restriction $|\omega\hat{p}_1(j\omega)|^{-1} \leqslant$ const. for $|\omega|>1$. Let $\|\cdot\|_W$ be the weighted seminorm, $\|P\|_W \triangleq \|\hat{w}P\|$, where $\hat{w}$ in H_o^∞ satisfies the condition $|\omega^\ell\hat{w}(j\omega)|^{-1} \leqslant$ const. for some integer ℓ. An optimal sequence of comparators Ω_n in $\mathbb{H}_o^\infty$ is sought. *A sequence with frequency responses*

$$\hat{q}_m(s) = c_m P_b^{-1}(s)[1-\hat{w}(\beta)\hat{w}^{-1}(s)](s+m)^{-\ell}(s+n_m)^{-2}, \quad m,n_m=1,2,\ldots \tag{5.1}$$

in which $c_m=m^\ell n_m^2$ *and*, *for each* m, n_m *is a sufficiently large integer*, *is such an optimal sequence*. *In fact*, $\mu(P_1)=\hat{w}(\beta)$, *and the sequence of sensitivities* $\|I-P_1 Q_m\|_W$ *approaches* $\mu(P_1)$ as $m\to\infty$.

6. MULTIVARIABLE SYSTEMS IN $\mathbb{H}^{\infty N}$

Multivariable systems are represented by operators $P \varepsilon \mathbb{H}^{\infty N}$, $P: H^{2N} \to H^{2N}$, with frequency response matrices $P = [\hat{p}_{ij}] \varepsilon H^{\infty N}$. Suppose that:
(a) P_a is a strongly causal operator in $\mathbb{H}_o^{\infty N}$, and
(b) P_b is in $\mathbb{H}^{\infty N}$. The frequency response $\hat{P}_b$ has a *minimal approximate inverse* $\hat{Q}_b$ in $(1+s)^\ell H^{\infty N}$; i.e., $\hat{w}\hat{P}_b\hat{Q}_b$ is in $H^{\infty N}$ and $\|\hat{w}(1-\hat{P}_b\hat{Q}_b)\|_{H^{\infty N}}=\mu(P_b)$.
Theorem 3 If (i) det $\hat{P}_a(s) \neq 0$ for Re(s) $\geqslant 0$ and (ii) there are constants $c > 0$, $\rho > 0$, and an integer $k > 0$ for which the inequality $|s|^k\sigma[\hat{P}_a(s)] \geqslant c$ is valid in the region $|s| > \rho$, Re(s) $\geqslant 0$, then, (a) $\mu(P_bP_a) = \mu(P_b)$, and (b) P_bP_a has an optimal sequence of approximate right inverses Q_m in $\mathbb{H}_o^{\infty N}$, with frequency responses
$\hat{Q}_m(s) \triangleq \hat{P}_a^{-1}(s)\hat{Q}_b(s)[m(s+m)^{-1}]^\ell[n_m(s+n_m)^{-1}]^{k+1}$
where for any $m=1,2,\ldots$, n_m is a sufficiently large integer.

Consider the feedback scheme of Eqs. 3.1 - 3.3. Suppose that the plant $P_1(s)$ is in $\mathbb{H}_o^{\infty N}$ and satisfies the restrictions: det $P_1(s) \neq 0$ for Re(s) $\geqslant 0$, and $|s|^k\sigma[\hat{P}_1(s)] > C$ in some region $|s| > p$, Re(s) $\geqslant 0$, where c,p are constants and k is an integer. The sensitivity $\eta \triangleq \|I-P_1Q\|_W$ is defined as in Sect. 5.
Corollary 6.2 The sensitivity $\|I-P_1Q_n\|_W$ can be made smaller than any $\varepsilon>0$ by a comparator Q_n in $\mathbb{H}_o^{\infty N}$ with frequency response, $Q_n(s) = P_1^{-1}(s)[n(s+n)^{-1}]^{k+1}$ n being a (sufficiently large) integer.

Let $\|\cdot\|_W$ be any weighted seminorm on $\mathbb{H}^{\infty N}$ of the form $\|P\|_W=\|\hat{w}\hat{P}\|$, $\hat{w}$ being (a scalar function) in H_o^∞. For any plant P in $\mathbb{H}^{\infty N}$, let the *rhp zeros* of P be points s_i in Re(s) $\geqslant 0$ $i=1,2,\ldots$, at which det $[\hat{P}(s_i)]=0$.
Theorem 4 $\mu(P) \geqslant \sup\{|w(s_i)|: i=1,2,\ldots\}$
Remark Theorem 4 implies that no feedback can produce small sensitivity if a plant zero is present in any heavily weighted part of the right half plane.

7. EFFECTS OF PLANT UNCERTAINTY ON DISTURBANCE ATTENUATION

We would like to study some effects of plant uncertainty on sensitivity to disturbances, and to show that at some point uncertainty reduces the possibility of constructing a feedback to attenuate disturbances.

Let $\mathbb{B}_r$ be a subspace of the causal operators $\mathbb{B}_s$. Consider the feedback (3.1) and model reference (3.3) schemes. Suppose that some nominal plant P_1 in $\mathbb{B}_r$ is specified and Eqs. 3.4 hold, so the two schemes are equivalent. The true plant P in $\mathbb{B}_r$ will differ from P_1 in general. Let $K_{33}: \mathcal{X} \to \mathcal{X}$, $K_{33}(d) \doteq y$ be the operator in $\mathbb{A}$ mapping disturbances into outputs. We have

$$K_{33} = (I-P_1Q)[I+(P-P_1)Q]^{-1} \tag{7.1}$$

Let $\mathbb{E}: \mathbb{A}_s^3 \to \mathbb{A}$, $\mathbb{E}(P,P_1,Q) = K_{33}$ be the function relating K_{33} to the model reference variables. Any pair (d,y) in $\mathcal{X}^2$ satisfying (3.1) or (3.3) also satisfies $y = \mathbb{E}(P,P_1,Q)d$.

The c.l. (closed-loop) operators K_{ij}, $i,j=2,3$ of the feedback scheme (3.1) were introduced in Sect. 3. Define the operators $K_{44} \triangleq [I+(P-P_1)Q]^{-1}$ and $K_{55} \triangleq [I+Q(P-P_1)]^{-1}$ and let K be the sextuplet of operators $\{K_{44},K_{55},P,P_1,Q,I\}$; the set $\{K_{ij}\}_{i,j=2}^{5}$ of all c.l. operators consists of algebraic combinations (i.e., involving sums and products only of) the operators in K; furthermore $K_{32} = -QK_{44}$.

Suppose a right seminorm $\|\cdot\|_r$ to be defined on $\mathbb{B}_r$ and recall that $\|\cdot\|_r \geqslant \|\cdot\|$. The true plant P will be supposed to lie in a *ball of uncertainty of radius $\delta \geqslant 0$ in $\mathbb{B}_r$ around the nominal* P_1, *$b(P_1,\delta) \triangleq \{P \text{ in } \mathbb{B}_r : \|P-P_1\|_r \leqslant \delta\}$*. Since $\|P-P_1\| \leqslant \|P-P_1\|_r$, $b(P_1,\delta)$ is a subset of the ball of δ in $\mathbb{B}$. An operator K in $\mathbb{A}$ which depends on $P \in \mathbb{B}_r$ will be called *bounded* over a ball $b(P_1,\delta)$ iff K is in $\mathbb{B}$ and there is a constant $c \geqslant 0$ with the property that $\|K\| \leqslant c$ for all P in $b(P_1,\delta)$. A feedback or model reference scheme will be *c.l. (closed-loop) bounded* on $b(P_1,\delta)$ iff all its c.l. operators are bounded on $b(P_1,\delta)$. The operators F in $\mathbb{A}_s$ and Q in $\mathbb{B}_s$ will be called *stabilizing* for $b(P_1,\delta)$ iff (3.1) and (3.3) respectively are c.l. bounded on $b(P_1,\delta)$.

Sensitivity

Suppose that right and left seminorms are defined on $\mathcal{B}$ as in Sect. 5, and induce a weighted seminorm $\|\cdot\|_W$ on $\mathbb{B}$. For the plant P in any ball of uncertainty $b(P_1,\delta)$, and any feedback F in $\mathbb{A}_s$ or comparator Q in $\mathbb{B}_s$, the *sensitivity to disturbances under plant uncertainty* of the equivalent schemes (3.1, 3.3) is defined to be

$$\eta(P_1,Q;\delta) = \sup_{P\varepsilon b(P_1,\delta)} \sup\{\|y\|_\ell/\|d\|_r : d \text{ in } \mathcal{B}_r,\ \|d\|_r \neq 0\}$$

If Q is (stabilizing) in $\mathbb{B}_s(P_1,\delta)$, and $\eta(P_1Q;\delta) = \infty$ otherwise. For any stabilizing Q,

$$\eta(P_1,Q;\delta) = \sup_{P\varepsilon b(P_1,\delta)} \|\mathbb{E}(P,P_1,Q)\|_W$$

by (7.2). We define the *optimal sensitivity to plant uncertainty to be*

$$\eta(P_1,\delta) = \inf_{Q\varepsilon\mathbb{B}_s} \sup_{P\varepsilon b(P_1,\delta)} \|\mathbb{E}(P,P_1,Q)\|_W$$

with the proviso that the sup is replaced by ∞ if Q is not stabilizing. For any fixed nominal plant P_1 in $\mathbb{B}_s$, $\eta(P_1,\delta)$ is a positive-real valued function of $\delta > 0$, which will be referred to as a *plant uncertainty tolerance function*.

Theorem 5 For any nominal plant P_1 in $\mathbb{B}_s$, the plant uncertainty tolerance function $\eta(P_1,\delta)$ is a monotone nondecreasing function of δ for $\delta \geqslant 0$; $\eta(P_1,\delta)$ approaches the singularity measure $\mu(P_1)$ as $\delta \to 0$; and $\eta(P_1,\delta) = \|I\|_W$ for $\delta = \|P_1\|_r$.

Remark Theorem 5 implies that whenever feedback reduces sensitivity for the nominal plant P_1, there are points in the interval $[0,\|P_1\|]$ at which $\eta(P_1,\delta)$ increases, i.e., where "the less we know about P, the less able we are to construct a feedback to attenuate disturbances".

8. FILTERING OF PLANT UNCERTAINTY: INVARIANT SCHEMES

Consider the flow graph of Fig. 4.

The new feedback scheme equations are,

$$y = Pv \quad (8.1a) \qquad\qquad v = Uu-Fy \quad (8.1b)$$

in which $P\varepsilon\mathbb{A}_s$, $F\varepsilon\mathbb{A}_s$ and $U\varepsilon\mathbb{A}$ are operators, and u,v, and y are in $\mathcal{X}$.

The c.l. operators are denoted by K_{ij}, $i,j=1,2,3$. The flow graph of Fig. 5 is described by the new model reference scheme equations,

$$y = Pv \qquad (8.3a) \qquad\qquad v = Ru-Q(y-P_1v) \qquad (8.3b)$$

in which P, $P_1 \varepsilon \mathbb{B}_s$, $R \varepsilon \mathbb{A}$, and $u,v,y \varepsilon \mathcal{X}$. For any $u \varepsilon \mathcal{X}$, Eqs. (8.3) have unique solutions for v and y in $\mathcal{X}$, as P, P_1, and Q are strictly causal. The schemes (8.1) and (8.3) will be called equivalent iff every triple (u,v,y) in $\mathcal{X}^3$ satisfying (8.1) satisfies (8.3) and vice versa. Equivalence will be established subject to the equations

$$Q = F(I+P_1F)^{-1} \qquad (8.4a) \qquad\qquad F = Q(I-P_1Q)^{-1} \qquad (8.4b)$$

$$R = (I+FP_1)^{-1}U \qquad (8.5a) \qquad\qquad U = (I+FP_1)R \qquad (8.5b)$$

For equivalent schemes, let $\mathbb{E} : \mathbb{A}_s^3 \to \mathbb{A}$, $\mathbb{E}(P,P_1,Q) = K_{33}$ be the function relating K_{33} to the model reference variables. Let $\mathbb{K}$ be the function $\mathbb{K} : \mathbb{A}_s^3 \times \mathbb{A} \to \mathbb{A}$, $\mathbb{K}(P,P_1,Q,R) = K_{13}$, which maps operators appearing in (8.3) into the c.l. operator K_{13}. Whenever all operators except P are regarded as fixed, $\mathbb{K}(P,P_1,Q,R)$ is denoted by $\mathbb{K}(P)$. For any P_1 in $\mathbb{A}_s$, $\mathbb{K}$ is called (nominal) plant invariant iff $\mathbb{K}(P_1) = P_1$; and, for any R_1 in $\mathbb{A}$, plant invariant $(\times R_1)$ iff $\mathbb{K}(P_1) = P_1R_1$.

From (8.4) it is clear that $\mathbb{K}$ has these properties whenever $R=I$ or $R=R_1$ respectively.

An operator $0_r \varepsilon \mathbb{B}$ is a right zero of P_1 iff $P_1 0_r = 0$.

Theorem 7 (a) Any feedback scheme (8.1) with (P,F,U) in $\mathbb{A}_s^2 \times \mathbb{A}$ is equivalent to a model reference scheme (8.3) with (P,P_1,Q,R) in $\mathbb{A}_s^3 \times \mathbb{A}$, in which (F,U) and (P_1,Q,R) are related by (8.4-8.5); and vice versa. If (8.4-8.5) hold, then: (b) If P assumes the (nominal) value $P_1 \varepsilon \mathbb{A}_s$, then $\mathbb{K}(P_1) = P_1R$, and if the feedback scheme is closed-loop stable, then Q and R of the model reference scheme are stable (although F and U may be unstable).
(c) The differences $\mathbb{K}(P) - \mathbb{K}(P_1)$ and $P-P_1$ are related by the formulas

$$\mathbb{K}(P) - \mathbb{K}(P_1) = (I-P_1Q)(P-P_1)\ \{I+Q(P-P_1)\}^{-1}R \qquad (8.6)$$

$$= \mathbb{E}(P,P_1Q)(P-P_1)R \qquad (8.7)$$

(d) For any $R_1 \varepsilon \mathbb{B}$, $\mathbb{K}$ is plant invariant $(\times R_1)$ iff $R = R_1+0_r$, where $0_r \varepsilon \mathbb{B}$ is any right zero of P_1.

Remarks Plant invariant schemes allow us to divide any control-law synthesis into two stages: (a) filtering of plant uncertainty $P-P_1$, and (b) design of a control law for a nominal plant P_1, with the assurance at least that the filtering stage will improve the design. In general, there is no "separation principle" to guarantee optimality of the division.

Either stage may come first. Therefore, in our theorems, P can be interpreted either as a plant without controller, for which a controller will be designed eventually; or, as a plant with controller attached, which requires additional filtering only to the extent that P differs from P_1.

Sensitivity to Plant Perturbations or Uncertainty

Suppose that a pair of seminorms $(\|\cdot\|_W, \|\cdot\|_1)$ is defined on the spaces $(\mathbb{B}, \mathbb{B}_r)$ respectively, where $\|\cdot\|_W$ is a left seminorm, and $\|\cdot\|_1 \leqslant \|\cdot\|_r$. A weighted seminorm $\|\cdot\|_V$ is assumed to be defined on the set of products $\mathbb{B} \cdot \mathbb{B}_r$, and to have the multiplicative property, $\|CD\|_V \leqslant \|C\|_W \|D\|_1$ for all C in $\mathbb{B}$ and D in $\mathbb{B}_r$. The pair $(\|\cdot\|_W, \|\cdot\|_1)$ will be called aligned iff there is a $D \varepsilon \mathbb{B}_r$ for which the preceeding inequality is an equality for all C in $\mathbb{B}$.

For the plant P in any ball of uncertainty $b(P_1,\delta)$, and any $(F \varepsilon \mathbb{A}_s$, $U \varepsilon \mathbb{A})$, or $(Q \varepsilon \mathbb{B}_s$, $R \varepsilon \mathbb{B})$, the sensitivity to plant perturbations (or uncertainty) of the equivalent schemes (8.1-8.3) is defined to be

$$\nu(P_1,Q,R;\delta) = \sup_{P \varepsilon b(P_1,\delta)} \{\|\mathbb{K}(P)-\mathbb{K}(P_1)\|_v \delta^{-1})\} \qquad (8.8)$$

for Q stabilizing (i.e., in $\mathbb{B}_s(P_1,\delta)$), and $\nu(P_1,Q,R,;\delta) = \infty$ otherwise. For any stabilizing Q, (8.7) gives the equation

$$\nu(P_1,Q,R;\delta) = \sup_{P \varepsilon b(P_1,\delta)} \{\|\mathbb{E}(P,P_1,Q)(P-P_1)R\|_v \delta^{-1}\} \qquad (8.9)$$

For any P_1 in $\mathbb{B}_s$ and $\delta > 0$, the optimal sensitivity to plant perturbations is defined to be

$$\nu(P_1,\delta) = \inf_{Q \varepsilon \mathbb{B}_s} \sup_{P \varepsilon b(P_1,\delta)} \{\|\mathbb{E}(P,P_1,Q)(P-P_1)\|_v \delta^{-1}\}$$

For fixed P_1, $\nu(P_1,\delta)$ is a function of $\delta \geqslant 0$ which will be referred to as the tolerance function of perturbation sensitivity to plant uncertainty.

Theorem 8 For any nominal plant P_1 in $\mathbb{B}_s$, the tolerance function $\nu(P_1,\delta)$ of perturbation sensitivity to plant uncertainty is a monotone nondecreasing function of $\delta \geqslant 0$, with the upper bounds $\nu(P_1,\delta) \leqslant \eta(P_1,\delta)$ and $\lim_{\delta\to 0}\nu(P_1,\delta) \leqslant \mu(P_1)$. If $\|\cdot\|_1 = \|\cdot\|_r$. and $(\|\cdot\|_W, \|\cdot\|_1)$ are aligned, then $\lim_{\delta\to 0}\nu(P_1,\delta) = \mu(P_1)$ and, for $\delta = \|P_1\|_r$, $\|P_1\|_v \|P_1\|_r^{-1} \leqslant \nu(P_1,\delta) \leqslant \|I\|_W$.

REFERENCES

[1] M. Horowitz, Synthesis of feedback systems, New York: Academic Press, 1963.

[2] G. Zames Nonlinear operators for system analysis, M.I.T., Res. Lab. of Electronics, Tech. Rept. 370, September 1960.

[3] C.A. Desoer, "Nonlinear distortion in feedback amplifiers" IRE Trans. on Circuit Theory, Vol. CT-9, pp. 2-6, March 1962.

[4] J.B. Cruz Jr., and W.R. Perkins, "A new approach to the sensitivity problem in multivariable feedback system design", IEEE Trans. Auto. Cont. AC-9, pp. 216-223, 1964.

[5] R.M. Desantis and W.A. Porter, "A generalized Nyquist plot and its use in sensitivity analysis", Int. J. Systems Sci., 1974, Vol. 5-12, pp. 1143-1153.

[6] G. Zames, "Feedback hierarchies and complexity", Decision and Control Conf., Dec. 1976.

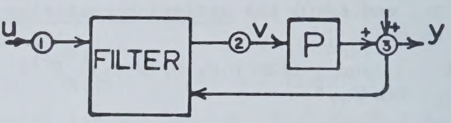

u
1
FILTER
2
v
P
+
d
+
3
y

FIG. 1

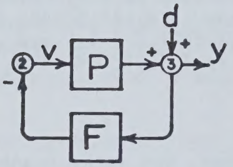


FIG. 2

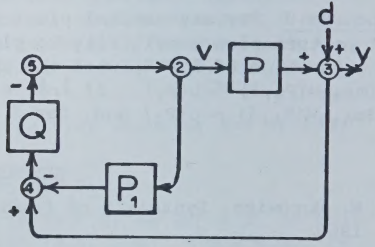

d
5
v
2
P
+
+
3
y
-
Q
4
-
P1
+

FIG. 3

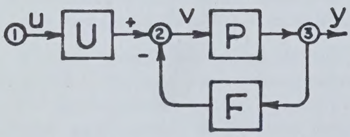


FIG. 4

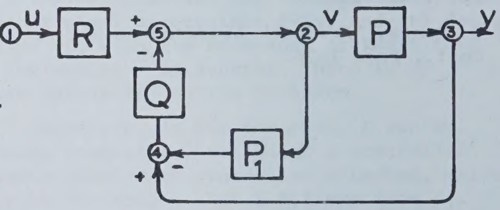

1
u
R
+
5
v
2
P
3
y
-
Q
4
-
P1
+

FIG. 5

LOOP CLOSURES AND THE INDUCED EXTERIOR MAP*

V. SESHADRI
Dept. Electrical Engineering
General Motors Institute
Flint, Michigan
USA 48502

M.K. SAIN
Dept. Electrical Engineering
University of Notre Dame
Notre Dame, Indiana
USA 46556

ABSTRACT

In this paper we establish the connection between linear multivariable feedback loop closures and the induced exterior map, the latter being the by-product of exterior algebras over the input and output vector spaces. This suggests the concept of a sequence of multivariable zeros which should enhance the designer's ability to shape multi-output transients.

Introduction

Much interest has been evinced in recent years about the exterior algebra, a structure especially suited for addressing questions related to matrix determinants and inverses. It has been shown that this structure can be used to solve problems in pole assignment [1,2], in individual zero placement [3] and indeed in a host of areas related to systems and information theory [4]. The present paper may roughly be divided into three parts. The first part introduces the exterior algebra. The presentation is extremely brief due to limitations of space; for more details the reader is referred to Greub [5]. The second part of the paper considers multivariable feedback loop closures, the loops being closed one by one, and studies their relation to certain entities called numerators of the kth kind, the latter arising from a frequency design method used in industry [6]. The last part of the paper establishes the connection between numerators of the kth kind and the kth induced exterior map, the latter being a by-product of exterior algebras over the input and output vector spaces [7]. In this way we will see that when considering multivariable feedback loop closures it would be helpful to view them in the context of the appropriate induced exterior morphism.

The Exterior Algebra

Consider an F-vector space V. We can construct an exterior algebra [5] ΛV over V. The bilinear operator introduced by this construction is commonly called the exterior product or the "wedge" product $\wedge$, and operates as

$$(\alpha_1 a_1 + \alpha_2 a_2) \wedge a_3 = \alpha_1 a_1 \wedge a_3 + \alpha_2 a_2 \wedge a_3$$

$$a_1 \wedge (\alpha_3 a_3 + \alpha_4 a_4) = \alpha_3 a_1 \wedge a_3 + \alpha_4 a_1 \wedge a_4,$$

where a_1, a_2, a_3, a_4 belong to the algebra ΛV, and $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ are field elements from F. Furthermore, the operator $\wedge$ is skew-symmetric.

Now consider a map $f : V \rightarrow W$. If we construct the exterior algebras ΛV and ΛW over the vector spaces V and W respectively, the map f induces a morphism $f^{\wedge}$ over the algebras [7], which is just a sequence of

---

*This work has been supported in part by the National Aeronautics and Space Administration under Grant NSG-3048 and in part by the Office of Naval Research under contract N00014-79-C-0475.

maps $f^\wedge_k$ over the kth exterior spaces, as shown in Figure 1.

$$\begin{tikzcd}
\Lambda V \arrow[d, "f^\wedge"] & : & F \arrow[d, "f^\wedge_o = 1"] & V \arrow[d, "f^\wedge_1 = f"] & \Lambda_2 V \arrow[d, "f^\wedge_2"] & \cdots & \Lambda_k V \arrow[d, "f^\wedge_k"] & \cdots \\
\Lambda W & & F & W & \Lambda_2 W & \cdots & \Lambda_k W & \cdots
\end{tikzcd}$$

Figure 1. The Induced Exterior Morphism, $f^\wedge$.

A reasonable question that could be asked at this juncture is what relevance the induced exterior morphism $f^\wedge$ has to multivariable feedback control design. The answer is that there already exists a feedback control design method in industry [6] that makes partial use of the induced exterior morphism structure. It is not clear, however, that the originators of the method are aware of the structure and it appears that the design method could be extended by making fuller use of the sequence of exterior maps. In order to address these issues, let us first consider a problem suggested by Hofmann, et al. [6] and study the connection between feedback loop closures and certain entities called numerators of various kinds, the latter arising when the following problem is solved.

Problem: Obtain expressions for the arbitrary closed-loop transfer function y_c/r_d after feeding back one or more outputs (measurements) to one or more inputs (controls), for the plant shown in Figure 2.

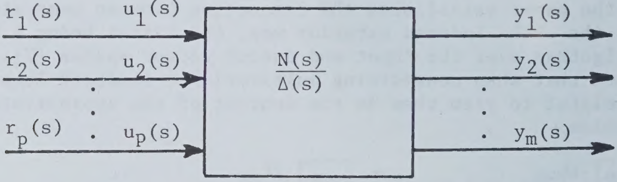


Figure 2. A Feedback Loop Closure Problem.

Comments: Figure 2 shows a multi-input multi-output plant; it has p inputs and m outputs. The reference, input and output may be compactly called r, u and y, and considered as elements of R(s)-vector spaces R, U and Y respectively, R(s) being the field of ratios of polynomials in s with real coefficients, the denominator being non-zero. The plant is nth order, linear and stationary. The expression for the plant in Figure 2, that is $N(s)/\Delta(s)$, is considered to have been derived, with the usual assumption of zero initial conditions, from an input-output plant description in the s-domain written as

$$A(s)\ y(s) = B(s)\ u(s),$$

where y belongs to the R(s)-vector space of dimension m, $[R(s)]^m$, and u belongs to $[R(s)]^p$. In the following discussion we shall drop the explicit dependence of the variables on s, for notational convenience.

The outputs y may be expressed explicitly in terms of the inputs u as

$$y = A^{-1} B u$$
$$= \frac{N}{\Delta} u ,$$

where Δ is the nth degree characteristic polynomial for the plant. If A and B are left coprime,

$$\Delta = \det A .$$

N is represented in numerical calculations by an $m \times p$ matrix whose elements belong to $R[s]$, that is, they are polynomials in s with real coefficients; N is the adjusted plant transfer function numerator matrix.

Solution: In the following treatment let the polynomial matrices A and B be expressed in terms of m-length column vectors as

$$A \overset{\Delta}{=} [a_1\ a_2\ \dots\ a_m] \quad , \quad B \overset{\Delta}{=} [b_1\ b_2\ \dots\ b_p].$$

Consider the case of a single feedback loop, that is, one output y_i is fed back to the jth comparison point through a feedback gain g_{ji}, as shown in Figure 3.

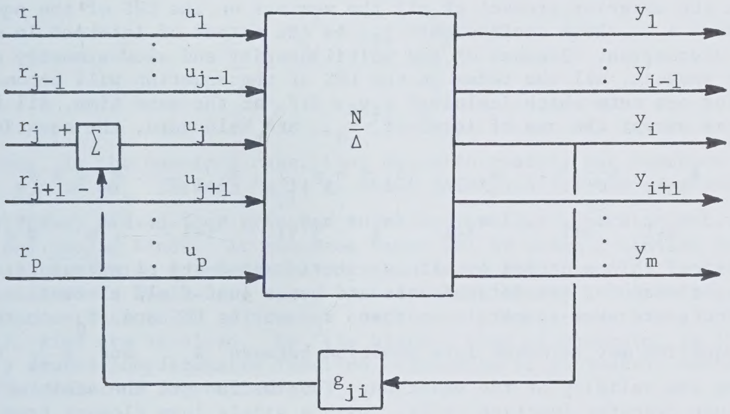


Figure 3. The Feedback Problem With One Loop Closure.

The relationship between the output y and the reference r may be expressed as

$$(A + BG) y = Br,$$

where y and u are given by

$$y = (y_1\ \dots\ y_m)^T \quad , \quad r = (r_1\ \dots\ r_p)^T .$$

G is a $p \times m$ matrix of zeros except for a single non-zero element g_{ji} ε $R(s)$, and is expressed in terms of P-length vectors in the same fashion as A and B above by

$$G = [0\ \dots\ 0\ g_i\ 0\ \dots\ 0]$$

where

$$g_i = (0\ ..\ 0\ g_{ji}\ 0\ ..\ 0)^T .$$

Note that g_{ji} is the jth element in g_i, and that g_i is the ith column of G. We can then express the equation

$$Ay = Bu$$

as

$$a_1 y_1 + .. + a_c y_c + .. + a_{i-1} y_{i-1} + (a_i + b_j g_{ji}) y_i + a_{i+1} y_{i+1} + .. + a_m y_m$$
$$= b_1 r_1 + .. + b_d r_d + .. + b_p r_p .$$

Note that in the above equation a_k, $k = 1,..,m$ and b_k, $k = 1,...,p$ are vectors whereas y_k, $k = 1,...,m$ and r_k, $k = 1,...,p$ are scalars, the field being $R(s)$.

We wish to isolate the closed-loop transfer function y_c/r_d; this may be achieved by taking the exterior product of both sides of the above equation with the (m-1)-exterior term

$$a_1 \wedge \cdots \wedge a_{c-1} \wedge a_{c+1} \wedge \cdots \wedge a_{i-1} \wedge (a_i + b_j g_{ji}) \wedge a_{i+1} \wedge \cdots \wedge a_m ,$$

which is the exterior product of all the vectors on the LHS of the equation except for a_c, whose coefficient y_c is the output of interest in our current discussion. Because of the multilinearity and skew-symmetry of the exterior product, all the terms on the LHS of the equation will become zero except for one term which includes $a_c y_c$. If, at the same time, all the references except the one of interest, r_d, are held zero, the equation becomes

$$a_1 \wedge \cdots \wedge a_{c-1} \wedge a_{c+1} \wedge \cdots \wedge a_{i-1} \wedge (a_i + b_j g_{ji}) \wedge a_{i+1} \wedge \cdots \wedge a_m \wedge a_c y_c$$
$$= a_1 \wedge \cdots \wedge a_{c-1} \wedge a_{c+1} \wedge \cdots \wedge a_{i-1} \wedge (a_i + b_j g_{ji}) \wedge a_{i+1} \wedge \cdots \wedge a_m \wedge b_d r_d .$$

Both sides of this equation contain m-exterior products of m-vectors; the products, therefore, are determinants and hence just field elements. Also, the products are skew-symmetric so that a_c on the LHS and b_d on the RHS of the equation may be moved into position between a_{c-1} and a_{c+1} while retaining the validity of the equation. Thus we can get the arbitrary closed-loop transfer function y_c/r_d with a single loop closure from the output y_i to the input u_j through the feedback element g_{ji}, as

$$\frac{y_c}{r_d} = \frac{a_1 \wedge \cdots \wedge a_{c-1} \wedge b_d \wedge a_{c+1} \wedge \cdots \wedge a_{i-1} \wedge (a_i + b_j g_{ji}) \wedge a_{i+1} \wedge \cdots \wedge a_m}{a_1 \wedge \cdots \wedge a_{c-1} \wedge a_c \wedge a_{c+1} \wedge \cdots \wedge a_{i-1} \wedge (a_i + b_j g_{ji}) \wedge a_{i+1} \wedge \cdots \wedge a_m}$$

or

$$\frac{y_c}{r_d} = \frac{\begin{array}{l}(a_1 \wedge \cdots \wedge a_{c-1} \wedge b_d \wedge a_{c+1} \wedge \cdots \wedge a_m \\ + a_1 \wedge \cdots \wedge a_{c-1} \wedge b_d \wedge a_{c+1} \wedge \cdots \wedge a_{i-1} \wedge b_j \wedge a_{i+1} \wedge \cdots \wedge a_m g_{ji})\end{array}}{(a_1 \wedge \cdots \wedge a_m + a_1 \wedge \cdots \wedge a_{i-1} \wedge b_j \wedge a_{i+1} \wedge \cdots \wedge a_m g_{ji})} .$$

In order to interpret the above expression for the closed-loop transfer function y_c/r_d observe that, in terms of numerical calculations, each of the two terms in the numerator and in the denominator of the above equation is simply a determinant; thus a total of four determinantal calculations is involved here. Determinants of this sort recur in transfer function expressions with one or more loops closed; they are called numerators of various kinds [6]. The kind of the numerator is dependent on the nature of the mix between the columns of A and the columns of B in the particular determinantal expression. For instance, the above equation contains two numerators of the first kind, that is, two determinants resulting from the m-exterior product of (m-1) columns of A with one column of B, namely

$$N_{u_d}^{y_c} = a_1 \wedge \cdots \wedge a_{c-1} \wedge b_d \wedge a_{c+1} \wedge \cdots \wedge a_m$$

and

$$N_{u_j}^{y_i} = a_1 \wedge \cdots \wedge a_{i-1} \wedge b_j \wedge a_{i+1} \wedge \cdots \wedge a_m .$$

Similarly, the equation has one numerator of the second kind, that is, the m-exterior product of (m-2) columns of A with two columns of B, namely

$$N_{u_d u_j}^{y_c y_i} = a_1 \wedge \cdots \wedge a_{c-1} \wedge b_d \wedge a_{c+1} \wedge \cdots \wedge a_{i-1} \wedge b_j \wedge a_{i+1} \wedge \cdots \wedge a_m .$$

Then, under the condition that $\det A = \Delta$, we can express the closed-loop transfer function y_c/r_d compactly as

$$\frac{y_c}{r_d} = \frac{N_{u_d}^{y_c} + g_{ji} N_{u_d u_j}^{y_c y_i}}{\Delta + g_{ji} N_{u_j}^{y_i}} .$$

Thus, in the one-loop case, that is, with exactly one feedback loop (from y_i to u_j through g_{ji} as shown in Figure 3), the expression for the arbitrary closed-loop transfer function involves numerators of the first and second kinds. It has been found [8] by using a similar exterior algebraic mechanism for manipulating the closed-loop transfer function expression that in the k-loop case, that is, with feedback loops from k different outputs to k different comparison points, numerators up to the (k+1)th kind are involved. Or, the highest kind of numerator in the arbitrary closed-loop transfer function expression is intimately connected with the number of loop closures.

The Numerator of the kth Kind and the kth Exterior Map

The arbitrary numerator of the kth kind is formed, as an extension of the definition of numerators of the first and second kind, by taking the m-exterior product of (m-k) columns of A with k columns of B, as

$$N_{u_{j_1} \cdots u_{j_k}}^{y_{i_1} \cdots y_{i_k}} = a_1 \wedge \cdots \wedge a_{i_1-1} \wedge b_{j_1} \wedge a_{i_1+1} \wedge \cdots \wedge a_{i_k-1} \wedge b_{j_k} \wedge a_{i_k+1} \wedge \cdots \wedge a_m .$$

We will show in the following that numerators of the kth kind may be understood in the context of the induced kth exterior map for the input R(s)-vector space U to the output R(s)-vector space Y. That, and the fact that numerators of different kinds are intimately related to the number of loop closures, would lead to the conclusion that there exists a strong association between the induced exterior map and loop closures.

Consider the equation relating the output y to the input u, expressed as

$$y = A^{-1} B u.$$

The vectors u and y belong to R(s)-vector spaces U and Y of dimension p and m respectively. Thus we have the morphism of vector spaces

$$A^{-1}B : U \to Y.$$

We have seen earlier that such a morphism of vector spaces induces a morphism of algebras

$$(A^{-1}B)^{\wedge} : \Lambda U \to \Lambda Y$$

as shown below

$$\begin{tikzcd}
\Lambda U \arrow[d, dashed, "(A^{-1}B)^{\wedge} :"] & R(s) \arrow[d, "1"] & U \arrow[d, "A^{-1}B"] & \Lambda_2 U \arrow[d, dashed, "(A^{-1}B)^{\wedge}_2"] & \cdot\,\cdot & \Lambda_k U \arrow[d, dashed, "(A^{-1}B)^{\wedge}_k"] & \cdot\,\cdot\,\cdot \\
\Lambda Y & R(s) & Y & \Lambda_2 Y & \cdot\,\cdot & \Lambda_k Y & \cdot\,\cdot\,\cdot
\end{tikzcd}$$

Figure 4. The Exterior Morphism Induced By $A^{-1}B$.

Of particular interest is the map

$$(A^{-1}B)^{\wedge}_k : \Lambda_k U \to \Lambda_k Y$$

because we intend to show how this map is related to numerators of the kth kind. In order to try and compute this map, we express [8]

$$(A^{-1}B)^{\wedge}_k = (A^{-1})^{\wedge}_k B^{\wedge}_k = (A^{\wedge}_k)^{-1} B^{\wedge}_k = \frac{\operatorname{adj} A^{\wedge}_k}{\det A^{\wedge}_k} \cdot B^{\wedge}_k .$$

Let us assume for the moment that we can find a matrix T such that

$$T A^{\wedge}_k = I \det A.$$

The reason for this assumption will become clear in the following; but for the moment, let us rewrite the above expression as

$$\frac{T}{\det A} \cdot A^{\wedge}_k = I.$$

Comparing the above with the definition of $\operatorname{adj} A^{\wedge}_k$ as

$$\frac{\operatorname{adj} A^{\wedge}_k}{\det A^{\wedge}_k} \cdot A^{\wedge}_k = I ,$$

and because $A^{\wedge}_k$ is invertible, we have

$$\frac{\operatorname{adj} A^{\wedge}_k}{\det A^{\wedge}_k} = \frac{T}{\det A} .$$

Thus we can replace $\operatorname{adj} A^{\wedge}_k/\det A^{\wedge}_k$ by $T/\det A$ in order to rewrite $(A^{-1}B)^{\wedge}_k$ as

$$(A^{-1}B)^{\wedge}_k = \frac{T B^{\wedge}_k}{\det A}.$$

The Nature of the Matrix T

We now investigate the nature of the matrix T; to understand T is to understand the nature of the kth exterior map $(A^{-1}B)^{\wedge}_k$, because $(A^{-1}B)^{\wedge}_k$ is just $TB^{\wedge}_k/\det A$. Recall that T has been defined in terms of its action on the image of $A_{\hat{k}}$, as

$$T A_{\hat{k}} = I \det A.$$

Now $A_{\hat{k}}$ is calculated by forming all possible kxk minors from the matrix A, and is thus

$$A_{\hat{k}} = \begin{bmatrix} A_{1..k,1..k} & .. & A_{1..k,i_1..i_k} & .. & A_{1..k,\overline{m-k+1..m}} \\ \vdots & & \vdots & & \vdots \\ A_{\overline{m-k+1..m},1..k} & .. & A_{\overline{m-k+1..m},i_1..i_k} & .. & A_{\overline{m-k+1..m},\overline{m-k+1..m}} \end{bmatrix}$$

where the overlines indicate grouping of terms, and the first and second set of subscripts of each element indicate the k rows and k columns, respectively, selected from the matrix A in order to form that particular kxk minor element. Expressing the matrix A as

$$A = [a_1 \;..\; a_{i_1-1} \; a_{i_1} \; a_{i_1+1} \;..\; a_{i_k-1} \; a_{i_k} \; a_{i_k+1} \;..\; a_m] \;,$$

the arbitrary ith column of $A_{\hat{k}}$ has been formed by selecting the k m-length columns $a_{i_1}, a_{i_2}, .., a_{i_k}$ from A and forming all possible kxk minors from these k columns by choosing k rows at a time according to some predetermined convention. Thus the arbitrary ith column of $A_{\hat{k}}$ is of length $\binom{m}{k}$, the number of combinations of k items chosen at a time from m.

Based on Laplace's Theorem [9], a possible construction for T would be to make the ith row of T consist of cofactors corresponding to the kxk minor elements in the arbitrary ith column of $A_{\hat{k}}$. Thus the elements of the ith row of T are formed from the (m-k) complementary columns $a_1, .., a_{i_1-1}, a_{i_1+1}, .., a_{i_k-1}, a_{i_k+1}, .., a_m$ of A. The (m-k) rows selected from these columns to form each element of the ith row T are complementary to the k rows selected from $a_{i_1}, .., a_{i_k}$ to form the corresponding element in the ith column of A. Hence the order in which the (m-k) rows of A are selected in forming the elements of T may be said to follow a complementary convention to that used in selecting k rows when forming $A_{\hat{k}}$. Hence the matrix T may be represented as

$$T = \begin{bmatrix} \tilde{A}_{\overline{k+1..m},\overline{k+1..m}} & . \; . \; . & \tilde{A}_{1..\overline{m-k},\overline{k+1..m}} \\ \vdots & & \vdots \\ \tilde{A}_{\overline{k+1..m},1..\overline{i_1-1}\;\overline{i_1+1..}\;\overline{i_k-1}\;\overline{i_k+1..m}} & . \; . \; . & \tilde{A}_{1..\overline{m-k},1..\overline{i_1-1}\;\overline{i_1+1..}\;\overline{i_k-1}\;\overline{i_k+1..m}} \\ \vdots & & \vdots \\ \tilde{A}_{\overline{k+1..m},1..\overline{m-k}} & . \; . \; . & \tilde{A}_{1..\overline{m-k},1..\overline{m-k}} \end{bmatrix} ,$$

the "tilde" symbol denoting that the minors are appropriately signed.

Recall that we are interested in the product $T B_{\hat{k}}$. The matrix $B_{\hat{k}}$ is formed from the columns of the matrix B in the same manner as $A_{\hat{k}}$ was from the columns of A. B consists of p columns of length m, as

$$B = [b_1 \;..\; b_{j_1-1} \; b_{j_1} \; b_{j_1+1} \;..\; b_{j_k-1} \; b_{j_k} \; b_{j_k+1} \;..\; b_p] \;,$$

and $B_{\hat{k}}$ is given as

$$B_{\hat{k}} = \begin{bmatrix} B_{1..k,1..k} & .. & B_{1..k,j_1..j_k} & .. & B_{1..k,\overline{p-k+1}..p} \\ \\ B_{\overline{m-k+1}..m,1..k} & .. & B_{\overline{m-k+1}..m,j_1..j_k} & .. & B_{\overline{m-k+1}..m,\overline{p-k+1}..p} \end{bmatrix}.$$

Here the arbitrary jth column of $B_{\hat{k}}$ is formed by picking the columns $b_{j_1},..,b_{j_k}$ from B and forming kxk minors using the same convention for the order in which k rows are chosen from m as before. Now consider a hybrid matrix

$$X_{ij} = [a_1 \; .. \; a_{i_1-1} \; b_{j_1} \; a_{i_1+1} \; .. \; a_{i_k-1} \; b_{j_k} \; a_{i_k+1} \; .. \; a_m]$$

and assume that we want to compute its determinant by invoking Laplace's Theorem. We may well select the k columns $b_{j_1},..b_{j_k}$ from X_{ij} and from kxk minors to be multiplied by the corresponding (m-k) x (m-k) cofactors from the remaining columns $a_1,..,a_{i_1-1},a_{i_1+1},..,a_{i_k-1},a_{i_k+1},..,a_m$ of X_{ij}. But the former kxk minors are identical to the elements in the arbitrary jth column of $B_{\hat{k}}$, and the latter (m-k) x (m-k) cofactors are identical to the elements in the arbitrary ith row of T. The order of selection of k rows to form the former elements and of the (m-k) rows to form the latter elements are precisely complementary by convention, and thus the product of the ith row of T with the jth column of $B_{\hat{k}}$, that is, the ijth element of $T\,B_{\hat{k}}$, is just $\det X_{ij}$, that is,

$$a_1 \wedge .. \wedge a_{i_1-1} \wedge b_{j_1} \wedge a_{i_1+1} \wedge .. \wedge a_{i_k-1} \wedge b_{j_k} \wedge a_{i_k+1} \wedge .. \wedge a_m .$$

But the above is precisely the arbitrary numerator of the kth kind

$$N^{y_{i_1}..y_{i_k}}_{u_{j_1}..u_{j_k}}$$

so that we have

$$(A^{-1}B)_{\hat{k}} = \frac{T\,B_{\hat{k}}}{\det A} = \begin{bmatrix} N^{y_1..y_k}_{u_1..u_k} & .. & N^{y_1..y_k}_{u_{j_1}..u_{j_k}} & .. & N^{y_1..y_k}_{u_{p-k+1}..u_p} \\ \vdots & & \vdots & & \vdots \\ N^{y_{i_1}..y_{i_k}}_{u_1..u_k} & .. & N^{y_{i_1}..y_{i_k}}_{u_{j_1}..u_{j_k}} & .. & N^{y_{i_1}..y_{i_k}}_{u_{p-k+1}..u_p} \\ \vdots & & \vdots & & \vdots \\ N^{y_{m-k+1}..y_m}_{u_1..u_k} & .. & N^{y_{m-k+1}..y_m}_{u_{j_1}..y_{j_k}} & .. & N^{y_{m-k+1}..y_m}_{u_{p-k+1}..u_p} \end{bmatrix} / \det A.$$

Hence the kth exterior map, represented in matrix form, has the numerators of the kth kind as its elements.

Conclusion

Whereas numerators of different kinds are used in closed-loop feedback design methods in industry via well-tested software packages, not much is available in the literature about their nature other than that they are determinants involving columns of A and B; we have seen in the above section, however, that they are, in fact, the elements of the appropriate induced exterior map. In earlier sections we saw the connection between feedback loop closures and numerators of different kinds; thus, as the feedback loops are closed one by one, the different induced exterior maps come into play. Current multivariable feedback design methods attempt to shape multioutput responses based only on the last exterior map and, consistent with this, present definitions of multivariable zeros involving only minors of the largest order in the transfer matrix. However, because the intermediate exterior maps are also involved when we close loops, it would seem that we would be making better use of the available information in shaping multi-output responses if we defined intermediate zeros from all the maps when we try to set up a valid multivariable counterpart of the familiar SISO zeros.

REFERENCES

1. V. Seshadri and M. Sain, "An Approach to Pole Assignment by Exterior Algebra," Proc. Fourteenth Allerton Conf. Circuit and System Thoery, pp. 399-407, 1976.

2. R.R. Gejji and M.K. Sain, "Application of Polynomial Techniques to Multivariable Control of Jet Engines," Proc. Fourth IFAC Symp. Multivariable Technological Systems, New Brunswick, Canada, 1977.

3. V. Seshadri and M.K. Sain, "An Application of Exterior Algebra to Multivariable Feedback Loops," Proc. 1979 Conf. Information Sciences and Systems, Baltimore, Maryland, pp. 337-342, 1979.

4. M.K. Sain, "The Growing Algebraic Presence in Systems Engineering: An Introduction," IEEE Proceedings, vol. 64, no. 1, pp. 96-111, 1976.

5. W.H. Greub, Multilinear Algebra, New York: Springer, 1967.

6. L.G. Hofmann, G.L. Teper and R.F. Whitbeck, "Application of Frequency Domain Multivariable Control Synthesis Techniques to an Illustrative Problem in Jet Engine Control," Intl. Forum Alternatives Linear Multivariable Control, Chicago, Illinois, 1977.

7. S. MacLane and G. Birkhoff, Algebra, London: Macmillan, 1967.

8. V. Seshadri, "Multivariable Feedback Loop Closures: An Approach by Exterior Algebra," Ph.D. dissertation, Department of Electrical Engineering, University of Notre Dame, Notre Dame, Indiana, October 1979.

9. F.E. Hohn, Elementary Matrix Algebra, New York: Macmillan, 1967.

ABSTRACT BILINEAR SYSTEMS: THE FORWARD SHIFT APPROACH

ARTHUR E. FRAZHO
University of Rochester
Department of Electrical Engineering
Rochester, New York 14627

ABSTRACT

A bilinear realization theory for a Volterra series input-output map is given. The approach involves a special transform representation for a Volterra series and certain shift operators on a Fock space.

1. INTRODUCTION

In this paper we present an abstract theory of discrete time bilinear systems. The foremost objective is to introduce a new mathematical structure that we feel is effective for dealing with these systems. The main ingredients are as follows: (i) a new transfer function for a Volterra-series input-output map; (ii) a "Fock space" $\rho(\mathcal{U})$ specially constructed to accommodate bilinear systems; (iii) certain shift operators S, T defined on $\rho(\mathcal{U})$ in such a way as to facilitate solutions to the bilinear realization problem.

There are a number of payoffs which results from handling bilinear systems in this new framework: having made the initial investment of creating the "right" coordinate free setting, the solution to the bilinear realization problem simply "falls out" (see Section 4). Using the shift operations S, T we obtain a restricted foreward shift realization (RFSR) for a Volterra series. The RFSR is a generalization to the bilinear setting of the Nerode type realizations used in linear systems [11]. That is our RFSR degenerates into this realization when the input-output map is linear. In addition our abstract setting also allows us to extend Helton's [9] controllability operator to the bilinear setting. The main advantage of this is the manner in which this operator is used to illuminate the connections between the RFSR and the concept of minimality. Finally, it is noted that the method presented here easily extends to state-affine or continuous time systems.

To discuss some of the work in a more mathematical setting some notation is established. Throughout $\mathcal{U}, \mathcal{V}, \mathcal{X},$ and $\mathcal{Y}$ are linear spaces over the field $\mathcal{K}$. These spaces can be finite or infinite-dimensional and no topological structure is needed. The space $\mathcal{U}$ [$\mathcal{Y}$] is fixed and always corresponds to inputs [outputs], respectively. The set of non-negative integers is denoted by $I \doteq \{n \geq 0\}$, and $I^n \doteq I \times I \times \ldots \times I$ denotes the n-fold Cartesian product of I. The space of all n-linear operator mapping $\mathcal{U}^n$ into $\mathcal{Y}$ is denoted by $\mathcal{L}(\mathcal{U}^n;\mathcal{Y})$. By convention $\mathcal{L}(\mathcal{U}^0;\mathcal{Y}) \doteq \mathcal{Y}$.

It is well known that discrete-time bilinear systems generate input-output maps of the following form:

$$y_n = \sum_{m=1}^{\infty} \sum_{i_m=0,\ i_{m-1}=0,\ \ldots,\ i_1=0}^{n-1,\ i_m-1,\ \ldots,\ i_2-1} \tilde{\theta}_m(i_2 - i_1, i_3 - i_2, \ldots, n - i_m)[u_{i_1}, u_{i_2}, \ldots, u_{i_m}] \tag{1.1}$$

where $u_k \varepsilon \mathcal{U}$ the input space, $y_k \varepsilon \mathcal{Y}$ the output space, and $\tilde{\Theta}_m(k_1, k_2, \ldots, k_m)$ is an element of $\mathcal{L}(\mathcal{U}^m; \mathcal{Y})$ for all $(k_1, k_2, \ldots, k_m) \varepsilon I^m$, $m > 0$. We call $\{\tilde{\Theta}_m\}_{m=1}^{\infty}$ the kernal sequence for system (1.1). The summation convention on $(i_1, i_2, \ldots, i_m)$ in (1.1) puts the kernals in strictly lower triangular form [1], [6], [10]. Due to this form we assume, without loss of generality, that

$$\left.\begin{array}{l} \tilde{\Theta}_m(k_1, k_2, \ldots, k_m) = 0 \\ \text{if one or more of the } k_i\text{'s are zero and } m > 0 \end{array}\right\} \tag{1.2}$$

The bilinear system we study is

$$x_{n+1} = Ax_n + N(u_n)x_n + Bu_n, \quad y_n = Cx_n \tag{1.3}$$

where $u_n \varepsilon \mathcal{U}$, $x_n \varepsilon \mathcal{X}$, $y_n \varepsilon \mathcal{Y}$ for all $n \geq 0$, A,B,C are linear operators on the appropriate spaces, and $N: \mathcal{U} \times \mathcal{X} \to \mathcal{X}$ is a bilinear operator. (For fixed $u \varepsilon \mathcal{U}$ $[x \varepsilon \mathcal{X}]$ $N(u)x$ is a linear operator in u[x], respectively.) System (1.3) is denoted by $\{A,B,C,N,\mathcal{X}\}$. The initial condition for (1.3) is always assumed to be zero, i.e., $x_0 = 0$. By recursively computing the solution for $\Sigma = \{A,B,C,N,\mathcal{X}\}$, it is easy to show that Σ generates an input-output map of the form (1.1), whose first two kernals are: $\tilde{\Theta}_{1\Sigma}(i + 1) = CA^iB$; $\tilde{\Theta}_{2\Sigma}(i + 1, j + 1)[u_1, u_2] = CA^jN(u_2)A^iBu_1$ for all $i \geq 0$, $j \geq 0$. The general term, denoted by $\tilde{\Theta}_{m\Sigma}$, is:

$$\tilde{\Theta}_{m\Sigma}(i_1 + 1, i_2 + 1, \ldots, i_m + 1)[u_1, u_2, \ldots, u_m] = \\ CA^{i_m}N(u_m)A^{i_{m-1}}N(u_{m-1}) \ldots N(u_2)A^{i_1}Bu_1 \tag{1.4}$$

where $(i_1, i_2, \ldots, i_m) \varepsilon I^m$, $m > 0$ and $u_i \varepsilon \mathcal{U}$. Due to equation (1.2) equation (1.4) completely determines the kernal sequence for Σ.

One of the problems we solve in this paper is the bilinear realization problem: given any input-output map of the form (1.1), (1.2) with kernal sequence $\{\tilde{\Theta}_m\}$, find a bilinear system Σ such that (1.1) is the input-output map for Σ. This amounts to finding a system Σ such that $\tilde{\Theta}_m = \tilde{\Theta}_{m\Sigma}$ for all $m>0$, see (1.4). If Σ solves the realization problem then Σ is called a realization of $\{\tilde{\Theta}_m\}$. The following standard terminology will be used. System $\Sigma = \{A,B,C,N,\mathcal{X}\}$ is finite dimensional if $\dim(\mathcal{X}) < \infty$. (The dimension of $\mathcal{X}$ is denoted by $\dim(\mathcal{X})$.) System Σ is a minimal realization of $\{\tilde{\Theta}_m\}$, if Σ is a realization of $\{\tilde{\Theta}_m\}$, and $\dim(\mathcal{X}) \leq \dim(\bar{\mathcal{X}})$ for all realizations $\bar{\Sigma} = \{\bar{A},\bar{B},\bar{C},\bar{N},\bar{\mathcal{X}}\}$ of $\{\tilde{\Theta}_m\}$. Let $\Sigma_i = \{A_i, B_i, C_i, N_i, \mathcal{X}_i\}$, $i = 1,2$ be two realizations of $\{\tilde{\Theta}_m\}$. If there exists a linear operator H mapping $\mathcal{X}_1$ into $\mathcal{X}_2$ such that $HA_1 = A_2H$; $HB_1 = B_2$; $C_2H = C_1$ and $HN_1(u) = N_2(u)H$ for all $u \varepsilon \mathcal{U}$, then H sends Σ_1 into Σ_2. If H is an isomorphism that sends Σ_1 into Σ_2, then Σ_1 is equivalent to Σ_2.

2. TRANSFORM NOTATION

In this section we define a transfer function for an input-output map of the form (1.1), (1.2). In particular the transfer function for the bilinear system $\Sigma = \{A,B,C,N,\mathcal{X}\}$ is given. The approach is similar to that presented in [5], [6].

First, some general notation is established. If $\mathcal{Q}$ is a subset of $\mathcal{X}$ then $\vee \mathcal{Q}$ is the (finite) linear span of $\mathcal{Q}$. If $\mathcal{V}_i$, $i > 0$ is a set of linear spaces then $\overset{\infty}{\underset{i=1}{\oplus}} \mathcal{V}_i \doteq \mathcal{V}_1 \oplus \mathcal{V}_2 \oplus \mathcal{V}_3 \oplus \ldots$ is the external direct sum of the $\mathcal{V}_i$'s. When the index starts at 1 and ends at ∞ it is dropped from the notation $\overset{\infty}{\underset{i=1}{\oplus}} \mathcal{V}_i$, i.e., $\overset{\infty}{\underset{i=1}{\oplus}} \mathcal{V}_i \doteq \oplus \mathcal{V}_i$ and $\oplus \mathcal{V}_{n+3} = \mathcal{V}_4 \oplus \mathcal{V}_5 \oplus \mathcal{V}_6 \oplus \ldots$ Elements in $\oplus \mathcal{V}_i$ are sometimes written as an infinite-touple. If $\oplus v_i \varepsilon \oplus \mathcal{V}_i$ then $\oplus v_i$ is written as $\oplus\; v_i = \{v_1, v_2, v_3, \ldots\}$. The notation $\oplus\; v_i$ and $\{v_1, v_2, v_3, \ldots\}$ is used interchangeably to represent the same element in $\oplus\, \mathcal{V}_i$.

Let $\tilde{f}$ be a sequence from I^n to the linear space $\mathcal{V}$. The formal power series f generated by $\tilde{f}$ is defined as:

$$f(\lambda_1, \lambda_2, \ldots, \lambda_n) = \sum_{i_1 \geq 0, i_2 \geq 0, \ldots, i_n \geq 0} \tilde{f}(i_1, i_2, \ldots, i_n) \lambda_1^{i_1} \lambda_2^{i_2} \ldots \lambda_n^{i_n} \tag{2.1}$$

where $\lambda_1, \lambda_2, \ldots, \lambda_n$ are the interminates. Since each $\tilde{f}(i_1, i_2, \ldots, i_n) \varepsilon \mathcal{V}$, the f generated by (2.1) is called a formal power series in $\lambda_1, \lambda_2, \ldots, \lambda_n$ with values in $\mathcal{V}$. The following convention is always used: a ~ over f always denotes the unique sequence $\tilde{f}$ that generates f, by (2.1). The set of all formal power series in $\lambda_1, \lambda_2, \ldots, \lambda_n$ with values in $\mathcal{V}$ is denoted by $\mathcal{S}_n(\mathcal{V})$. The set of all polynomials (finite sums in (2.1)) in $\lambda_1, \lambda_2, \ldots, \lambda_n$ with values in $\mathcal{V}$ is denoted by $\mathcal{P}_n(\mathcal{V})$. Clearly $\mathcal{P}_n(\mathcal{V})$, $\mathcal{S}_n(\mathcal{V})$ are linear spaces over the field $\mathcal{K}$, and $\mathcal{P}_n(\mathcal{V}) \subseteq \mathcal{S}_n(\mathcal{V})$.

Consider the n-th kernal $\tilde{\Theta}_n$ in (1.1). For each $n > 0$ we define the modified transfer function M_n for $\tilde{\Theta}_n$ by the following formal series:

$$M_n(\lambda_1, \lambda_2, \ldots, \lambda_n) \doteq \sum_{i_1=0, i_2=0, \ldots, i_n=0}^{\infty, \infty, \ldots, \infty} \tilde{\Theta}_n(i_1+1, i_2+1, \ldots, i_n+1) \lambda_1^{i_1} \lambda_2^{i_2} \ldots \lambda_n^{i_n} \; . \tag{2.2}$$

Clearly M_n is an element in $\mathcal{S}_n(\mathcal{L}(\mathcal{U}^n;\mathcal{Y}))$. M_n is called the modified transfer for $\tilde{\Theta}_n$ because $\lambda_1 \lambda_2 \ldots \lambda_n M_n = \Theta_n$ is the usual transfer function for $\tilde{\Theta}_n$ [5], [6]. Note that Θ_n is generated by $\tilde{\Theta}_n$, see (2.1). The modified transfer function M for system (1.1), or for the kernal sequence $\{\tilde{\Theta}_n\}_{n=1}^{\infty}$, is

defined as the external direct sum of the M_n's, i.e., $M \doteq \oplus M_n \doteq \overset{\infty}{\underset{n=1}{\oplus}} M_n$ where M_n is defined in (2.2). Consulting (1.2) it is clear that $M = \oplus M_n$ and the sequence $\{\tilde{\Theta}_n\}_{n=1}^{\infty}$ uniquely determine each other. It is important to note that the modified transfer function M is an element of the following linear space:

$$\mathcal{S} \doteq \mathcal{S}(\mathcal{U};\mathcal{Y}) = \overset{\infty}{\underset{n=1}{\oplus}} \mathcal{S}_n(\mathcal{L}(\mathcal{U}^n;\mathcal{Y})) \; . \tag{2.3}$$

The n-th component of $\mathcal{S}$ is a formal power series in $\lambda_1, \lambda_2, \ldots, \lambda_n$ with values in $\mathcal{L}(\mathcal{U}^n;\mathcal{Y})$.

From (1.4), $\Sigma = \{A, B, C, N, \mathcal{X}\}$ generates an input-output map of the form (1.1). Using (2.2) and (1.4) a simple calculation shows that the modified transfer function M_Σ for Σ is: $M_\Sigma = \oplus M_{n\Sigma}$ where the first two components are $M_{1\Sigma}(\lambda_1) = CF_1B$; $M_{2\Sigma}(\lambda_1, \lambda_2)[u_1, u_2] = CF_2N(u_2)F_1Bu_1$. The general term is

$$M_{n\Sigma}(\lambda_1, \lambda_2, \ldots, \lambda_n)[u_1, u_2, \ldots, u_n] = CF_nN(u_n)F_{n-1}N(u_{n-1}) \ldots N(u_2)F_1Bu_1 \tag{2.4}$$

where $u_i \varepsilon \mathcal{U}$ and F_m is the formal power series defined by

$$F_m \doteq \sum_{i \geq 0} \lambda_m^i A^i \doteq (I - \lambda_m A)^{-1} . \tag{2.5}$$

The modified transfer function for Σ is always denoted by $M_\Sigma \doteq \oplus M_{n\Sigma}$. Clearly $M_\Sigma \varepsilon \mathcal{S}$, i.e., the n-component $M_{n\Sigma}$ of M_Σ is a formal series in $\mathcal{S}_n(\mathcal{L}(\mathcal{U}^n;\mathcal{Y}))$. Thus in the transform domain the bilinear realization problem becomes: Given any M in $\mathcal{S}$ find a bilinear system such that $M_\Sigma = M$.

3. Tensors, Operators and Spaces

In this section, we introduce several operators and linear spaces that will be used to solve the bilinear realization problem. Throughout, tensor product spaces will play an important role. The linear space formed by the n-fold tensor product of $\mathcal{U}$ is denoted by $\otimes\mathcal{U}^n \doteq \mathcal{U}\otimes\mathcal{U}\otimes\dots\otimes\mathcal{U}$ ($\mathcal{U}$ appears n-times). By convention $\otimes\mathcal{U}^1 \doteq \mathcal{U}$. For each H in $\mathcal{L}(\mathcal{U}^n;\mathcal{Y})$ we define the linear operator $H\otimes$ mapping $\otimes\mathcal{U}^n$ into $\mathcal{Y}$ by

$$H\otimes u_1\otimes u_2\otimes\dots\otimes u_n \doteq H[u_1,u_2,\dots,u_n] \tag{3.1}$$

where $u_1\otimes u_2\otimes\dots\otimes u_n \in \otimes\mathcal{U}^n$.

A linear space of utmost importance is

$$\mathcal{P} \doteq \mathcal{P}(\mathcal{U}) \doteq \overset{\text{finite}}{\underset{n=1}{\oplus}} \mathcal{P}_n(\otimes\mathcal{U}^n) \ . \tag{3.2}$$

The finite direct sum in (3.2) indicates that all elements in $\mathcal{P}$ have compact support, i.e., $\overset{\infty}{\underset{n=1}{\oplus}} p_n \doteq \{p_1,p_2,p_3,\dots\}\in\mathcal{P}$ if and only if the n-th component p_n is a polynomial in $\lambda_1,\lambda_2,\dots,\lambda_n$ with values in $\otimes\mathcal{U}^n$, and $p_k = 0$ for all k sufficiently large. The operator E embedding $\mathcal{U}$ in $\mathcal{P}$ is defined by

$$Eu \doteq \{u,0,\dots,0,0,\dots\} \tag{3.3}$$

where $u\in\mathcal{U}$. The shift operator S mapping $\mathcal{P}$ into $\mathcal{P}$ is defined by

$$S\oplus p_n \doteq \oplus\,\lambda_n\,p_n \qquad (\oplus p_n \in \mathcal{P}) \ . \tag{3.4}$$

The linear operator S multiplies the n-th component, p_n of $\oplus p_n$, by λ_n. The bilinear operator T mapping $\mathcal{U}\times\mathcal{P}$ into $\mathcal{P}$ is defined by

$$T(u)\oplus p_n \doteq \{0,p_1\otimes u,\ p_2\otimes u,\ p_3\otimes u,\ \dots\} \tag{3.5}$$

where $u\in\mathcal{U}$ and $\oplus p_n \in \mathcal{P}$. Finally, for each M in $\mathcal{S}$ we define the operator $M\otimes$ mapping $\mathcal{P}$ into $\mathcal{Y}$ by

$$M\otimes\oplus p_n \doteq$$

$$\sum_{n\geq 1}\ \sum_{i_1\geq 0,i_2\geq 0,\dots,i_n\geq 0} \tilde{M}_n(i_1,i_2,\dots,i_n)\otimes\tilde{p}_n(i_1,i_2,\dots,i_n) \tag{3.6}$$

where $\oplus p_n \in \mathcal{P}$, $M = \oplus M_n$, and $\tilde{M}_n$, $\tilde{p}_n$ generate the formal power series M_n, p_n, respectively, see (2.1). Since the elements in $\mathcal{P}$ have compact support $M\otimes$ is a well defined linear operator.

4. The Forward Shift Realizations

In this section, the realization problem is solved. The idea behind our approach is to use shift operators to obtain a realization of M. Then use an equivalence relation to extract a minimal realization. By using the shift operators presented in Section 3, it is easy to construct a realization of M. Consulting (2.1), (2.4) and (2.5), $\Sigma = \{A,B,C,N,\mathcal{X}\}$ is a realization of $M = \oplus M_n$ if

$$\tilde{M}_n(i_1,i_2,\dots,i_n)[u_1,u_2,\dots,u_n]=CA^{i_n}N(u_n)A^{i_{n-1}}N(u_{n-1})\dots N(u_2)A^{i_1}Bu_1 \tag{4.1}$$

for all $(i_1, i_2, \ldots, i_n) \varepsilon I^n$, all $n>0$ and all $u_i \varepsilon \mathcal{U}$. (Recall that $\tilde{M}_n$ generates M_n, by (2.1).) One set of operators that satisfy (4.1) is $A = S$, $B = E$, $C = M\otimes$ and $N = T$. To verify, this is almost trivial. For instance, when $n = 3$:

$$M \otimes S^k T(u_3) S^j T(u_2) S^i E u_1 = M \otimes S^k T(u_3) S^j T(u_2) \{\lambda_1^i u_1, 0, 0, \ldots\} =$$
$$M \otimes S^k T(u_3) \{0, \lambda_1^i \lambda_2^j u_1 \otimes u_2, 0, 0, \ldots\} = \tag{4.2}$$
$$M \otimes \{0, 0, \lambda_1^i \lambda_2^j \lambda_3^k u_1 \otimes u_2 \otimes u_3, 0, 0, \ldots\} = M_3(i,j,k)[u_1, u_2, u_3].$$

Hence (4.1). The general case follows in a similar manner giving

Proposition (4.1). If $M \varepsilon \mathcal{S}$ then $\Gamma \doteq \{S, E, M\otimes, T, \mathcal{P}\}$ is a realization of M.

System Γ is called the forward shift realization (FSR) of M. The state space for the FSR of M is "too big." In fact the same state-space $\mathcal{P}$ is used to realize all modified transfer functions in $\mathcal{S}$. To obtain a smaller state-space, consider the following linear subspace $\mathcal{P}_{\bar{o}}$ of $\mathcal{P}$:

$$\mathcal{P}_{\bar{o}} \doteq \cap\{\mathcal{N}(M \otimes S^{i_n} T(u_n) S^{i_{n-1}} T(u_{n-1}) \ldots T(u_2) S^{i_1} \mid (i_1, i_2, \ldots, i_n) \varepsilon I^n, \quad n>0 \text{ and } u_k \varepsilon \mathcal{U}\}. \tag{4.3}$$

($\mathcal{N}$ denotes the null space.) Let $\mathcal{P}_o$ be the quotient space of $\mathcal{P}$ modulo $\mathcal{P}_{\bar{o}}$; it is denoted by $\mathcal{P}_o \doteq \mathcal{P}/\mathcal{P}_{\bar{o}}$. A typical element in $\mathcal{P}_o$ is the coset $\{p_o + \mathcal{P}_{\bar{o}}\}$ where $p_o \varepsilon \mathcal{P}$. Since $\mathcal{P}_{\bar{o}}$ is an invariant subspace for S and all $T(u)$, $u \varepsilon \mathcal{U}$ the following operators are well defined:

$$\begin{aligned} S_o &: \mathcal{P}_o \to \mathcal{P}_o, \; S_o\{p_o + \mathcal{P}_{\bar{o}}\} \doteq \{Sp_o + \mathcal{P}_{\bar{o}}\}, \\ E_o &: \mathcal{U} \to \mathcal{P}_o, \; Eu \doteq \{Eu + \mathcal{P}_{\bar{o}}\} \qquad (u \varepsilon \mathcal{U}), \\ M_o \otimes &: \mathcal{P}_o \to \mathcal{Y}, \; M_o \otimes \{p_o + \mathcal{P}_{\bar{o}}\} \doteq M \otimes p_o, \\ T_o &: \mathcal{U} \times \mathcal{P}_o \to \mathcal{P}_o, \; T_o(u)\{p_o + \mathcal{P}_{\bar{o}}\} = \{T(u)p_o + \mathcal{P}_{\bar{o}}\} \quad (u \varepsilon \mathcal{U}). \end{aligned} \tag{4.4}$$

Using (4.1) to (4.4) it is easy to verify

Proposition (4.2). If $M \varepsilon \mathcal{S}$ then $\Gamma_o \doteq \{S_o, E_o, M_o \otimes, T_o, \mathcal{P}_o\}$ is a realization of M.

System Γ_o is called the restricted forward shift realization (RFSR) of M. The space $\mathcal{P}_o$ degenerates into the "Nerode" or "natural" state space of [11], [4] when the input-output map (1.1) is linear, homogeneous. Thus the RFSR is a generalization of these realizations. Note we did not use a Nerode equivalence relation or a module set up to obtain $\mathcal{P}_o$. The state-space $\mathcal{P}_o$ simply falls out from the FSR and (4.3).

5. Minimal Realizations

In this section, the RFSR is shown to be a minimal realization. As a corollary to this we obtain many of the standard results on minimality for bilinear systems. The approach depends upon the reachability operator $K_\Sigma \otimes$.

Let $\Sigma = \{A,B,C,N,\mathcal{X}\}$ and $\mathcal{S}(\mathcal{U};\mathcal{X})$ be the space defined in (2.3) with $\mathcal{Y}$ replaced by $\mathcal{X}$. The element K_Σ in $\mathcal{S}(\mathcal{U};\mathcal{X})$ is defined by $K_\Sigma \doteq \oplus K_{n\Sigma}$ where the first two terms are $K_{1\Sigma}(\lambda_1) = F_1B$; $K_{2\Sigma}(\lambda_1,\lambda_2)[u_1,u_2] = F_2N(u_2)F_1Bu_1$. The general term is

$$K_{n\Sigma}(\lambda_1,\lambda_2,\ldots,\lambda_n)[u_1,u_2,\ldots,u_n] = F_nN(u_n)F_{n-1}N(u_{n-1})\ldots N(u_2)F_1Bu_1 \tag{5.1}$$

where $u_k \varepsilon \mathcal{U}$ and F_m is given (2.5). The operator $K_\Sigma \otimes$ mapping $\mathcal{P}$ into $\mathcal{X}$ (see (3.6)) is called the reachability operator for Σ. For the linear version of this operator, see [9]. An important property of $K_\Sigma \otimes$ is given in

Lemma (5.1). If Σ is a realization of M then $K_\Sigma \otimes$ sends the FSR of M into Σ.

Proof. The proof requires the verification of the appropriate identities. Let $\Sigma = \{A,B,C,N,\mathcal{X}\}$. Consulting (2.4) and using $M = M_\Sigma$ it is clear that $CK_\Sigma = M_\Sigma = M$ or $CK_\Sigma \otimes = M\otimes$. From (5.1) and the definition of E; $K_\Sigma \otimes E = B$. Now let $f_n = \{0,0,\ldots,0,\lambda_1^{i_1}\lambda_2^{i_2}\ldots\lambda_n^{i_n}u_1 \otimes u_2 \otimes \ldots \otimes u_n,0,0,\ldots\}$ where the nonzero component appears in the n-th position and $u_i \varepsilon \mathcal{U}$, then:

$$AK_\Sigma \otimes f_n = AA^{i_n}N(u_n)A^{i_{n-1}}N(u_{n-1})\ldots N(u_2)A^{i_1}Bu_1 =$$

$$K_\Sigma \otimes \lambda_n f_n = K_\Sigma \otimes Sf_n \ .$$

Since elements of the form of f_n span $\mathcal{P}$, $AK_\Sigma \otimes = K_\Sigma \otimes S$. Finally, using the same f_n it can be shown that $N(u)K_\Sigma \otimes f_n = K_\Sigma \otimes T(u)f_n$ for all $u\varepsilon \mathcal{U}$. Hence, $N(u)K_\Sigma \otimes = K_\Sigma \otimes T(u)$. #

Corollary (5.1). If Σ is a realization of $M\varepsilon\mathcal{S}$ then $\mathcal{N}(K_\Sigma\otimes) \subseteq \mathcal{P}_{\bar{o}}$ where $\mathcal{P}_{\bar{o}}$ is defined in (4.3).

Proof. Let $\Sigma = \{A,B,C,N,\mathcal{X}\}$ and $p\varepsilon\mathcal{N}(K_\Sigma\otimes)$. Clearly

$$CA^{i_n}N(u_n)A^{i_{n-1}}N(u_{n-1})\ldots N(u_2)A^{i_1}K_\Sigma \otimes p = 0 \tag{5.2}$$

for all $(i_1,i_2,\ldots,i_n)\varepsilon I^n$, all $n>0$ and all $u_k\varepsilon\mathcal{U}$. Lemma (5.1) and the definition of sends into implies that the $K_\Sigma\otimes$ in (5.2) can be moved to obtain:

$$CK_\Sigma \otimes S^{i_n}T(u_n)S^{i_{n-1}}T(u_{n-1})\ldots S(u_2)S^{i_1}p = 0 \ . \tag{5.3}$$

The definition of sends into gives $CK_\Sigma\otimes = M\otimes$. Therefore by (5.3) $p\varepsilon\mathcal{P}_{\bar{o}}$ and $\mathcal{N}(K_\Sigma\otimes) \subseteq \mathcal{P}_{\bar{o}}$. #

Let Σ be a realization of M. Corollary (5.1) gives

$$\dim(\mathcal{P}_o) = \dim(\mathcal{P}/\mathcal{P}_{\bar{o}}) \le \dim(\mathcal{P}/\mathcal{N}(K_\Sigma\otimes)) = \dim(\mathcal{R}(K_\Sigma\otimes)) \le \dim(\mathcal{X}) \ . \tag{5.4}$$

Recalling that $\mathcal{P}_o$ is the state-space for the RFSR proves

Proposition (5.1). Let $M \varepsilon \mathcal{S}$. The RFSR of M is a minimal realization of M. In particular M admits a finite-dimensional bilinear realization if and only if the dimension of $\mathcal{P}_o$ is finite.

For all systems Σ', $\mathcal{N}(K_\Sigma,\otimes) \subseteq \mathcal{P}_{\bar{o}}$. For those special bilinear systems $\Sigma = \{A,B,C,N,\mathcal{X}\}$ where $\mathcal{N}(K_\Sigma\otimes) = \mathcal{P}_{\bar{o}}$, we define another reachability operator, $K_{o\Sigma}\otimes$ mapping $\mathcal{P}_o$ into $\mathcal{V}$ by: $K_{o\Sigma}\otimes\{p_o + \mathcal{P}_{\bar{o}}\} \doteq K_\Sigma \otimes p_o$ where $\{p_o + \mathcal{P}_{\bar{o}}\}$ is an element of $\mathcal{P}_o$. (Whenever we write $K_{o\Sigma}\otimes$ it is always understood that it is well defined, i.e., $\mathcal{N}(K_\Sigma\otimes) = \mathcal{P}_{\bar{o}}$.) Clearly $K_{o\Sigma}\otimes$ is one-to-one. A simple calculation with Lemma (5.1) verifies that $K_{o\Sigma}\otimes$ sends the RFSR of M into Σ. We are now ready to prove

Proposition (5.2). Let M admit a finite-dimensional realization. If Σ is a minimal realization then Σ is equivalent to the RFSR of M. In fact, $K_{o\Sigma}\otimes$ is an isomorphism that sends the RFSR of M into Σ.

Proof. The minimality of Σ and (5.4) gives $\dim(\mathcal{X}) = \dim(\mathcal{P}_o)$. Using Corollary (5.1)

$$\dim(\mathcal{P}_o) = \dim(\mathcal{P}/\mathcal{P}_{\bar{o}}) \leq \dim(\mathcal{P}/\mathcal{P}_{\bar{o}}) + \dim(\mathcal{P}_{\bar{o}}/\mathcal{N}(K_\Sigma\otimes))$$

$$= \dim(\mathcal{P}/\mathcal{N}(K_\Sigma\otimes)) = \dim(\mathcal{R}(K_\Sigma\otimes)) \leq \dim(\mathcal{X}) = \dim(\mathcal{P}_o) < \infty \qquad (5.5)$$

($\mathcal{R}$ denotes the range of an operator.) Therefore, all inequalities are equalities and $\dim(\mathcal{P}_{\bar{o}}/\mathcal{N}(K_\Sigma\otimes)) = 0$ or $\mathcal{P}_{\bar{o}} = \mathcal{N}(K_\Sigma\otimes)$. Thus $K_{o\Sigma}\otimes$ is well defined. Because $K_{o\Sigma}\otimes$ is one-to-one and $\dim(\mathcal{X}) = \dim(\mathcal{P}_o)$, $K_{o\Sigma}\otimes$ is an isomorphism. Since $K_{o\Sigma}\otimes$ sends the RFSR into Σ the proof is complete. #

The transitivity of system equivalence yields

Corollary (5.2). Let M admit a finite-dimensional realization. Any two minimal realizations of M are equivalent.

In this section, we have developed a theory of minimality without using the concepts of reachability and observability. In the next section we develop a theory of reachability and observability without using the concept of minimality.

6. Reachability and Observability

We begin with

Definition (6.1). Let $\Sigma = \{A,B,C,N,\mathcal{X}\}$, (i) System Σ is span reachable if $\mathcal{X} = \mathcal{R}(K_\Sigma\otimes)$, or equivalently if

$$\mathcal{X} = \vee\{A^{i_n}N(u_n)A^{i_{n-1}}N(u_{n-1})\ldots N(u_2)A^{i_1}Bu_1 \mid n>0,$$
$$(i_1,i_2,\ldots,i_n)\varepsilon I^n, \text{ and } u_k \varepsilon \mathcal{U}\}. \qquad (6.1)$$

(The name reachibility operator comes from the fact that Σ is span reachable if and only if $K_\Sigma\otimes$ is onto.) (ii) System Σ is observable if

$$\{0\} = \cap\{\mathcal{N}(CA^{i_n}N(u_n)A^{i_{n-1}}N(u_{n-1})\ldots N(u_2)A^{i_1})\mid$$
$$(i_1,i_2,\ldots,i_n)\varepsilon I^n, n>0 \text{ and } u_k \varepsilon \mathcal{U}\}. \qquad (6.2)$$

Clearly the FSR is span reachable, i.e., (6.1) holds. Using (4.4), (6.1) and the reachability of the FSR it is almost trivial to verify that the RFSR is span reachable. The definition of $\mathcal{P}_{\bar{o}}$ (4.3) guarantees that the RFSR is observable. Thus,

Proposition (6.1). The RFSR of M is a span reachable and observable realization of M.

Proposition (6.2). If Σ is a span reachable and observable realization of M, then Σ is equivalent to the RFSR of M. In fact, $K_{o\Sigma} \otimes$ is an isomorphism that sends the RFSR of M into Σ.

Proof. From Corollary (5.1) $\mathcal{N}(K_\Sigma \otimes) \subseteq \mathcal{P}_{\bar{o}}$. Using the observability of Σ we show that we actually have equality. If $p \varepsilon \mathcal{P}_{\bar{o}}$, then for all $(i_1, \dots, i_n) \varepsilon I^n$, all $n > 0$ and all $u_k \varepsilon \mathcal{U}$ the following holds:

$$M \otimes S^{i_n} T(u_n) S^{i_{n-1}} T(u_{n-1}) \dots T(u_2) S^{i_1} p = 0 \tag{6.3}$$

By Lemma (5.1) $M \otimes = CK_\Sigma \otimes$ or (6.3) becomes

$$CK_\Sigma \otimes S^{i_n} T(u_n) S^{i_{n-1}} T(u_{n-1}) \dots T(u_2) S^{i_1} p = 0. \tag{6.4}$$

The definition of sends into allows us to move the $K_\Sigma \otimes$ in (6.4) to get $CA^{i_n} N(u_n) A^{i_{n-1}} N(u_{n-1}) \dots N(u_2) A^{i_1} K_\Sigma \otimes p = 0$ for all $(i_1, i_2, \dots, i_n) \varepsilon I^n$, all $n > 0$ and all $u_k \varepsilon \mathcal{U}$ But Σ is observable so $K_\Sigma \otimes p = 0$. Hence $\mathcal{P}_{\bar{o}} \subseteq \mathcal{N}(K_\Sigma \otimes)$, or by Corollary (5.1) $\mathcal{P}_{\bar{o}} = \mathcal{N}(K_\Sigma \otimes)$.

Clearly $K_{o\Sigma} \otimes$ is well defined, one to one and sends the RFSR into Σ. Because Σ is span reachable $K_{o\Sigma} \otimes$ is onto: By Definition (6.1) $\mathcal{V} = \mathcal{R}(K_\Sigma \otimes) = \mathcal{R}(K_{o\Sigma} \otimes)$. Thus $K_{o\Sigma} \otimes$ is an isomorphism and the proof is complete. #

Using the span reachability, observability and minimality of the RFSR, Propositions (5.2) and (6.2) give the following standard results [1], [10] [12]:

Corollary (6.2a). All span reachable and observable realizations of M are equivalent.

Corollary (6.2b). Let M admit a finite-dimensional realization. System Σ is a minimal realization of M if and only if Σ is a span reachable and observable realization of M.

Note that Proposition (6.2) and Corollary (6.2a) hold for finite and infinite-dimensional bilinear systems. So these results are slightly more general than those in the literature.

7. Conclusion

In this paper, we have used shift operators on the "Fock space" $\mathcal{P}$ to solve the bilinear realization problem. The idea was to use the operators S and T to obtain a span reachable realization of M, called the FSR. Then use the space $\mathcal{P}_{\bar{o}}$ to extract a span reachable and observable realization of M, called the RFSR.

The operators S, T and the space $\mathcal{P}$ are believed to be new. However, the idea of using shift operators to solve a realization problem is not new. Shift operators have been used to develop a theory of linear systems [7], [9], [11]. We have generalized these systems to the bilinear setting, i.e., the RFSR of M degenerates into these realizations when $M = M_1 \otimes 0 \otimes 0 \otimes 0 \dots$ In the nonlinear setting shift operators have also been used to solve certain realization questions. A particular kind of a column shift on a (behavior) Hankel matrix appears in [12] to solve the state-affine realization problem.

In [5] and [6] the backward shift operator is used to develop a theory of bilinear systems. The backward shift approach of [5][6] is the "dual" of the foreward approach presented here. The transfer function M used here degenerates into the regular transfer function of [2][3] when the input-output map is homogeneous. For other aspects of bilinear systems see [1][10].

References

[1] C. Bruni, G. DiPillo and G. Koch. Bilinear systems: an appealing class of nearly linear systems in theory and applications. IEEE Trans. Automatic Control, AC-19 (1974), pp. 334-348.

[2] S.J. Clancy and W.J. Rugh. On the realization problem for stationary, homogeneous discrete-time systems, Automatica, Vol. 14, pp. 357-366, 1978.

[3] S.J. Clancy, G.E. Mitzel and W.J. Rugh. On transfer function representations for homogeneous nonlinear systems, IEEE Trans. on Automatic Control, to appear.

[4] E. Fornasini and G. Marchesini. Algebraic realization theory of bilinear discrete-time input-output maps. J. Franklin Inst., (1976), pp. 143-159.

[5] A.E. Frazho. Shift operators and bilinear system theory. Proc. of the 1978 Conference on Decision and Control, pp. 551-556.

[6] A.E. Frazho. A shift operator approach to bilinear system theory, submitted.

[7] P.A. Fuhrman. On realizations of linear systems and applications to some questions of stability. Math. Systems Theory, 8 (1974), pp. 132-141.

[8] E.G. Gilbert. Functional expansions for the response of nonlinear differential systems. IEEE Trans. Automatic Control, AC-22 (1977), pp. 909-921.

[9] J.W. Helton. Discrete time systems, operator models, and scattering theory. J. Functional Analysis, 16 (1974), pp. 15-38.

[10] A. Isidori and A. Ruberti. Realization theory of bilinear systems. Geometric Methods in System Theory, Ed. D.Q. Mayne and R.W. Brockett, D. Reidel Publishing Co., Dordrecht, 1973.

[11] R.E. Kalman, P.L. Falb and M.A. Arbib, Topics in Mathematical System Theory. McGraw-Hill, New York, 1969.

[12] E.D. Sontag. Realization theory of discrete-time nonlinear systems: Part I - The bounded case. IEEE Transactions on Circuits and Systems, Vol. CAS 26, No. 4 (1979), pp. 342-356.

TEAM-OPTIMAL CLOSED-LOOP STACKELBERG STRATEGIES IN LINEAR-QUADRATIC DIFFERENTIAL GAMES

TAMER BASAR[1)]
Department of Applied Mathematics
Twente University of Technology
Enschede, The Netherlands

GEERT JAN OLSDER[2)*]
Division of Applied Sciences
Harvard University
Cambridge, MA 02138

ABSTRACT

Closed-loop Stackelberg (CLS) solutions of linear quadratic continuous time differential games are determined. Explicit conditions are obtained under which the CLS solution is a representation of the optimal feedback solution of a related team problem which is defined as the joint minimization of the leader's cost function.

1. INTRODUCTION

In this paper the closed-loop Stackelberg (CLS) solution of two-player nonzero-sum linear quadratic differential games is obtained, under certain, not totally restrictive, conditions on the system parameters. Both the finite-and infinite-horizon problems are considered, and in both cases the derived CLS strategies have the important property that they generate the same state trajectory as that of a related optimal team problem. The objective function of this team problem is precisely the cost function of the leader, so that the derived CLS solution can also be considered as an extreme Pareto-optimal solution to the leader's advantage.

In the derivation of the CLS solutions that inherit the above property, two different classes of representations have been adopted for the leader: (i) linear strategies that depend only on the current and initial values of the state, (ii) linear strategies that depend on the whole past trajectory. For the former class of representations, both the finite and infinite-horizon problem are solved, with conditions of existence related to solutions of linear matrix equations. The scalar case has been worked out in detail, and it has been shown that, within that content, one of the following two situations can arise: (i) the differential game admits a CLS solution within the prescribed class, (ii) there exist only ε-Stackelberg strategies for the leader with the corresponding sequence of costs converging to the minimum value of his cost function. For the class of strategies that depend on the whole past trajectory, a set of sufficiency conditions that are related to the solutions of two linear matrix differential equations are obtained, under which the finite-horizon problem admits a CLS solution with the above cited feature. It has been conjectured, however, that this more general class of representations does not possess any superiority on the former class of strategies as far as the sufficiency conditions are concerned, and this claim has been verified for the scalar case.

In (1) the same problem has been considered with respect to linear-quadratic difference games. The specific techniques used in the current paper are inherently different than those in (1) and lead to explicit and computationally attractive expressions. In (2), where also the same problem has been considered, the leader uses linear memory strategies described by Lebesgue-Stieltjes measures. Within that general setting the sufficiency conditions obtained are difficult to manipulate.

---

1) On leave from Maramara Research Institute, Gebze, Kocaeli, Turkey
2) On leave from Twente University of Technology, Enschede, The Netherlands

2. GENERAL PROBLEM STATEMENT AND SOLUTION OF THE RELATED TEAM PROBLEM.

Consider a system of which the evolution of the state is described by the differential equation

$$\frac{dx}{dt} = Ax + B_1u + B_2v, \; x(0) = x_o \; , \; 0\leq t\leq T \; , \qquad (2.1)$$

where $u(\cdot)$ is an r-vector function controlled by Player 1 (the leader), $v(\cdot)$ is an r'-vector function controlled by Player 2 (the follower), $x(\cdot)$ is the state vector of dimension n, A, B_1 and B_2 are matrices of appropriate dimensions. The terminal time T will be assumed, for the moment, to be a finite number. The cost function for Player 1 and 2 are given by J_1 and J_2, respectively, where

$$J_i = \frac{1}{2} x'(T)Q_{if}x(T) + \frac{1}{2}\int_o^T \{x'Q_ix + u'R_{i1}u + v'R_{i2}v\}\, dt \qquad (2.2)$$

with $R_{11}>0$, $R_{22}>0$, $R_{12}>0$, all other weighting matrices being nonnegative definite. The symbol ' denotes the transpose operation. All this information and the value of the initial state x_o are known to both players. Each player has access to closed-loop perfect state information and can utilize that in the choice of his control. We denote typical strategies for the leader and the follower by γ_1 and γ_2, respectively, whose realizations are denoted above by u and v, respectively; that is $u(t) = \gamma_1(t,x(s),s\leq t)$ and $v(t) = \gamma_2(t,x(s),s\leq t)$. We further assume that γ_1 and γ_2 satisfy the standard measurability, continuity and growth conditions so that the differential equation admits a unique solution on [o,T], for each pair (γ_1,γ_2). We denote the space of such admissible strategies for Player i by Γ_i, i = 1,2. Finally, we denote the value of J_i given by (2.2) for an admissible pair of strategies (γ_1,γ_2) by $J_i(\gamma_1,\gamma_2)$.

Definition 1.

For the differential game formulated above, a strategy $\gamma_1^* \in \Gamma_1$ constitutes a closed-loop Stackelberg (CLS) strategy for the leader if

$$\sup_{R(\gamma_1^*)} J_1(\gamma_1^*,\gamma_2) \leq \sup_{R(\gamma_1)} J_1(\gamma_1,\gamma_2) \; , \; \forall\, \gamma_1 \in \Gamma_1, \qquad (2.3)$$

where $R(\gamma_1)$ denotes the rational reaction set of the follower defined by

$$R(\gamma_1) = \{\gamma_2^\circ \in \Gamma_2 \colon J_2(\gamma_1,\gamma_2^\circ) \leq J_2(\gamma_1,\gamma_2),\ \forall\, \gamma_2 \in \Gamma_2\} \qquad (2.4)$$

The CLS cost of the leader is then

$$J_1^* = \sup_{R(\gamma_1^*)} J_1(\gamma_1^*,\gamma_2) = \inf_{\Gamma_1} \sup_{R(\gamma_1)} J_1(\gamma_1,\gamma_2). \qquad (2.5)$$

Furthermore, any strategy $\gamma_2^* \in R(\gamma_1^*)$ for which $J_1(\gamma_1^*,\gamma_2^*) = J_1^*$ is known as a CLS strategy for the follower. □

We note that the natural bound

$$J_1^* \geq \min_{\Gamma_1} \min_{\Gamma_2} J_1(\gamma_1,\gamma_2) = J_1(\tilde{\gamma}_1,\tilde{\gamma}_2) \overset{\Delta}{=} \tilde{J}_1 \qquad (2.6)$$

obtained by jointly minimizing $J_1(\gamma_1,\gamma_2)$ over Γ_1 and Γ_2, is the best the leader can hope to accomplish. The RHS of (2.6) defines a team problem that admits a well-defined solution, since $J_1(\gamma_1,\gamma_2)$ is a strictly convex functional of its arguments. Its unique solution in feedback form is

$$\tilde{\gamma}_i\,(t,x(t)) = -\,R_{1i}^{-1}B_i'S(t)x(t)\ ,\ i = 1,2 \tag{2.7}$$

with $S(\cdot)$ defined as the unique nonnegative-definite matrix solution of

$$\dot{S} = -\,Q_1 - A'S - SA + S[B_1R_{11}^{-1}B_1' + B_2R_{12}^{-1}B_2']S, S(T) = Q_{1f} \tag{2.8}$$

The unique trajectory resulting from application of strategies (2.7) is

$$\overline{x}(t) = \Phi(t,o)x_o \tag{2.9}$$

where $\Phi(t,\tau)$ is the state transition matrix of the system

$\dot{x} = [A - (B_1R_{11}^{-1}B_1' + B_2R_{12}^{-1}B_2')S]\ x.$ Let us now consider a subset of Γ_1

consisting of all strategies for the leader with open-loop representation identical to $\tilde{\gamma}_1(t,\overline{x}(t))$, that is, introduce a set $\tilde{\Gamma}_1 \subset \Gamma_1$ defined as

$$\tilde{\Gamma}_1 = \{\gamma_1 \in \Gamma_1 : \gamma_1(t,x(s),s \leq t) = -\,R_{11}^{-1}B_1'S(t)\Phi(t,0)x_o\} \tag{2.10}$$

Analogously, define a set $\tilde{\Gamma}_2 \subset \Gamma_2$ for the follower:

$$\tilde{\Gamma}_2 = \{\gamma_2 \in \Gamma_2 : \gamma_2(t,x(s),s \leq t) = -\,R_{12}^{-1}B_2'S(t)\Phi(t,0)x_o\} \tag{2.11}$$

Our interest in these sets stems from the following property:

Proposition 1.

If there exists a strategy $\gamma_1^* \in \tilde{\Gamma}_1$ such that all solutions of the minimization problem

$$\inf_{\Gamma_2} J_2(\gamma_1^*,\gamma_2) \tag{2.12}$$

lie in $\tilde{\Gamma}_2$, then γ_1^* constitutes a CLS strategy for the leader and $\tilde{\gamma}_2$ is a CLS strategy for the follower. □

In the remainder of the paper we will investigate to what extent the hypothesis of Proposition 1 is satisfied. We will restrict ourselves to an even smaller subset of $\tilde{\Gamma}_1$, comprised of strategies of the form

$$\gamma_1(t,x(t),x_o) = -\,R_{11}^{-1}B_1'S(t)x(t) + P'(t)[x(t) - \overline{x}(t)] \tag{2.13}$$

which depend at time t only on $x(t)$ and x_o. Later, in section 6, we will briefly discuss the implications of a broader class of strategies.

3. SUFFICIENT CONDITIONS FOR THE LEADER TO ENFORCE HIS TEAM SOLUTION WITHIN A SPECIFIC REPRESENTATION CLASS.

Assuming that the leader chooses a strategy in the form (2.13), where $P'(\cdot)$ is an $(r\times n)$-dimensional matrix whose entries are continuous functions on $[0,T]$, then the follower's reaction will be determined from the solution of the following LQ optimal control problem:

$$\min_{\gamma_2 \in \Gamma_2} [\tfrac{1}{2} x'(T) Q_{2f} x(T) + \tfrac{1}{2} \int_o^T [x'\{Q_2 + (P' - R_{11}^{-1}B_1'S)'R_{21}(P' - R_{11}^{-1}B_1'S)\}x - 2x'(P' - R_{11}^{-1}B_1'S)'R_{21}P'\overline{x} + u_2'R_{22}u_2]\,dt] \tag{3.1}$$

subject to

$$\frac{dx}{dt} = (A - B_1R_{11}^{-1}B_1'S + B_1P')x - B_1P'\overline{x} + B_2u_2, x(o) = x_o \tag{3.2}$$

and with

$$u_2(t) = \gamma_2(t, x(s), s \leq t). \tag{3.3}$$

We adopt the Hamiltonian approach, and define the Hamiltonian H corresponding to (3.1) and (3.2) as

$$H = \tfrac{1}{2} x'\{Q_2 + (P' - R_{11}^{-1}B_1'S)'R_{21}(P' - R_{11}^{-1}B_1'S)\}x - x'(P' - R_{11}^{-1}B_1'S)'R_{21}P'\overline{x} + \tfrac{1}{2}u_2'R_{22}u_2 + \mu'(A - B_1R_{11}^{-1}B_1'S + B_1P')x - \mu'B_1P'\overline{x} + \mu'B_2u_2, \tag{3.4}$$

where $\mu(\cdot)$ denotes the n-dimensional costate vector, satisfying the differential equation

$$\dot{\mu} = -\{Q_2 + (P' - R_{11}^{-1}B_1'S)'R_{21}(P' - R_{11}^{-1}B_1'S)\}x + (P' - R_{11}^{-1}B_1'S)'R_{21}P'\overline{x} - (A - B_1R_{11}^{-1}B_1'S + B_1P')'\mu; \quad \mu(T) = Q_{2f}\, x(T). \tag{3.5}$$

Maximization of H with respect to u_2 yields

$$u_2(t) = -R_{22}^{-1}B_2'\mu(t), \; t \in [0,T], \tag{3.6}$$

and this has to be matched, in view of (3.5), with any element of $\tilde{\Gamma}_2$, in particular with the open-loop strategy.

Let us now assume, for the moment, that there exists a matrix function $P(\cdot)$ such that the hypothesis of Proposition 1 is valid. Then, equating (3.6) with $\tilde{\gamma}_2(t, \overline{x}(t))$, we obtain the relation

$$R_{22}^{-1}B_2'\mu(t) = R_{12}^{-1}B_2'S(t)\overline{x}(t), \; t \in [0.T], \tag{3.7}$$

and furthermore, since the optimal trajectories of the team problem and the present optimal control problem will coincide, we can let $x(\cdot) \equiv \overline{x}(\cdot)$ in (3.5) to yield a differential equation whose solution depends linearly on $\overline{x}(\cdot)$, which in turn is related to x_o through an invertable transformation. Therefore, letting $\mu(t) = M(t)\overline{x}(t)$ where $M(\cdot)$ is an $(n \times n)$ matrix function whose entries are continuously differentiable on $[0,T]$, we readily obtain the following linear matrix differential equation for M to satisfy

$$\dot{M} + MF + F'M + PB_1'M - PR_{21}R_{11}^{-1}B_1'S + Q_2 + SB_1R_{11}^{-1}R_{21}R_{11}^{-1}B_1'S + SB_2R_{12}^{-1}B_2'M = 0 \; ; \; M(T) = Q_{2f}, \tag{3.8}$$

where $F \triangleq A - (B_1R_{11}^{-1}B_1' + B_2R_{12}^{-1}B_2')S$. An equivalent relation replacing (3.7) is now

$$R_{22}^{-1}B_2'M(t) = R_{12}^{-1}B_2'S(t). \tag{3.9}$$

Condition 1.

There exists an $(n \times r)$ matrix function $P(\cdot)$ with continuous entries so that the unique solution of the linear matrix differential equation (3.8) satisfies the relation (3.9), where $S \geq 0$ is defined as the unique solution of (2.8) and the team trajectory $\bar{x}(\cdot)$ is given by (2.9). □

Theorem 1.

Let Condition 1 be satisfied for a matrix function $P^*(\cdot)$. Then, there exists a CLS solution for the differential game of section 2, which is given by

$$\gamma_1^*(t,x(t),x_o) = [P^{*\prime}(t) - R_{11}^{-1}B_1'S(t)]\, x(t) - P^{*\prime}(t)\bar{x}(t), \tag{3.10a}$$

$$\gamma_2^*(t,x(t)) = - R_{12}^{-1}B_2'S(t)x(t), \tag{3.10b}$$

and the CLS cost for the leader and the follower are given respectively by

$$J_1^* = \frac{1}{2}\, x_o'S(0)x_o, \tag{3.11a}$$

$$J_2^* = \frac{1}{2}\, x_o'M(0)x_o. \quad \square \tag{3.11b}$$

4. INVESTIGATION OF THE SCALAR CASE

In this section, we investigate the implication of Theorem 1 for the case when all matrices and vectors are of dimension one and, in addition, $Q_{1f} = Q_{2f} = 0$. In particular, we illustrate that, with the exception of one singular case with respect to the parameters of the game, Theorem 1 provides the complete solution to this scalar version of the Stackelberg problem of section 2.

Let us first assume that $B_1 \neq 0$ and $B_2 \neq 0$. Then, by appropriate rescaling, the scalar version can be formulated with the state equation (2.1) replaced by

$$\dot{x} = ax + u + v,\ x(0) = x_o\ ,\ o \leq t \leq T. \tag{4.1}$$

Furthermore, the cost functions (2.2) can be taken as

$$J_i = \frac{1}{2} \int_o^T \{q_i x^2 + r_{i1}u^2 + v^2\}\, dt \tag{4.2}$$

with $r_{11} > 0$, $r_{21} \geq 0$, $q_i \geq 0$. In terms of this new notation, Condition 1 is equivalent to existence of a continuous function $p(\cdot)$ that satisfies the equation

$$p(r_{11} - r_{21}) = r_{11}(q_1 - q_2) + (r_{11} - r_{21})s^2\ , \tag{4.3}$$

where s satisfies

$$\dot{s} + 2as + q_1 - (1 + 1/r_{11})s^2 = 0,\ s(T) = 0. \tag{4.4}$$

Eq. (4.3) clearly admits a unique p-solution provided $r_{11} \neq r_{21}$. Hence, we immediately have the following result as a direct application of Theorem 1 to the scalar problem.

Corollary 1.

If $r_{11} \neq r_{21}$ then the scalar differential game problem, as defined by

(4.1) and (4.2), admits a CLS solution with $J_1^* = \tilde{J}_1$. □

If $r_{11} = r_{21}$ and $q_1 = q_2$ then all possible choices of p satisfy eq. (4.3). This is, of course, as it should be expected, since then $J_1 = J_2$, and thus any representation of the optimal team solution is, trivially, a CLS strategy, with $J_1^* = \tilde{J}_1$, for the leader.

If $r_{11} = r_{21}$ and $q_1 \neq q_2$, we show below that, in such a case, there exist ε-Stackelberg strategies for the leader. To this end, we first give a precise definition of an ε-Stackelberg strategy:

Definition 2.

For the differential game problem formulated in section 2, a strategy $\gamma_1^\varepsilon \in \Gamma_1$ is said to constitute an ε-Stackelberg strategy for the leader if

$$\sup_{R(\gamma_1^\varepsilon)} J_1(\gamma_1^\varepsilon, \gamma_2) - J_1^* < \varepsilon, \tag{4.5}$$

where ε is some positive number and J_1^* denotes the CLS cost for the leader as defined by (2.5). □

Proposition 2.

Let $r_{11} = r_{21}$ for the scalar differential game of this section. Then, given an $\varepsilon > 0$, there exists an ε-Stackelberg strategy for the leader, and as $\varepsilon \downarrow 0$ his ε-Stackelberg cost converges to his optimal team cost $\tilde{J}_1$.

Proof: In the formulation of the scalar differential game, replace r_{11} by $r^\delta = r_{11} + \delta$, where $\delta > 0$ is a constant. Denote the minimum value of the related team problem by $\tilde{J}_1(\delta)$. $\tilde{J}_1(\delta)$ is clearly a continuously differentiable and increasing function of δ, with the property $\tilde{J}_1(\delta) \downarrow \tilde{J}_1$ as $\delta \downarrow 0$, where $\tilde{J}_1 = \tilde{J}_1(0)$ is the optimal team cost related to the original problem. It, then, follows that, given an $\varepsilon > 0$, one can find a $\delta(\varepsilon)$ such that $\tilde{J}_1(\delta) - \tilde{J}_1 < \varepsilon$.

Now for the new differential game with r_{11} replaced by r^δ, the leader's strategy is denoted by γ_1^ε and he clearly achieves a cost of $\tilde{J}_1(\delta)$. In the original game, the actual cost incurred to the leader will be $\tilde{J}_1(\delta) - \delta\int_0^T [\gamma_1^\varepsilon]^2\, dt$. Since $J_1^* \geq \tilde{J}_1$ and $\tilde{J}_1(\delta) - \tilde{J}_1 < \varepsilon$, Propos. 2 follows. □

5. THE INFINITE - HORIZON PROBLEM

We now consider the closed-loop Stackelberg solution of the differential game problem of section 2 for the case when $T \to \infty$. We let $Q_{1f} = Q_{2f} = 0$. Since this infinite-horizon problem can be considered as the limiting case of the finite-horizon differential game, versions of Condition 1 and Theorem 1 can easily be obtained by letting the derivative terms vanish in the differential equations for S and M, but provided that the pair $\{A, (B_1, B_2)\}$ is stabilizable, a condition which we assume to hold a priori. The statement of Proposition 1 is equally valid in this case with some obvious notational modifications, as well as the approach taken in section 3 to obtain a candidate Stackelberg solution for the leader. Hence, adopting the representation

$$\gamma_1(t, x(t), x_o) = -R_{11}^{-1}B_1'Sx(t) + P'[x(t) - \bar{x}(t)], \tag{5.1}$$

for the leader, where P is an $(n\times r)$ matrix yet to be determined and where $S > 0$ is defined as the unique solution of

$$S[B_1R_{11}^{-1}B_1' + B_2R_{12}^{-1}B_2']\,S - SA - A'S - Q_1 = 0, \tag{5.2}$$

we seek to solve the infinite-horizon optimization problem faced by the follower. By going through similar steps as in section 3, it is now not difficult to visualize the following versions of Condition 1 and Theorem 1, which require no further justification:

Condition 2.

There exists an element P such that

(i) $\{A - B_1R_{11}^{-1}B_1'S + B_1P, B_2\}$ is stabilizable; (5.3)

(ii) the linear matrix equation

$$MF + F'M + PB_1'M - PR_{21}R_{11}^{-1}B_1'S + Q_2 + S[B_1R_{11}^{-1}R_{21}R_{11}^{-1}B_1 + B_2R_{12}^{-1}R_{22}R_{12}^{-1}B_2]S=0, \tag{5.4}$$

where

$$F \triangleq A - (B_1R_{11}^{-1}B_1' + B_2R_{12}^{-1}B_2')S \tag{5.5}$$

has a unique solution M which satisfies the constraint

$$R_{22}^{-1}B_2'M = R_{12}^{-1}B_2'S. \qquad \square \tag{5.6}$$

Theorem 2.

Let Condition 2 be satisfied for a specific matrix P^*. Then, there exists a CLS solution to the infinite-horizon LQ differential game, which is given by

$$\gamma_1^*(t,x(t),x_o) = [P^{*\prime} - R_{11}^{-1}B_1'S]x(t) - P^{*\prime}\overline{x}(t) , \tag{5.7a}$$

$$\gamma_2^*(x(t)) \quad = \quad - R_{12}^{-1}B_2'Sx(t). \qquad \square \tag{5.7b}$$

A special class of problems of particular interest are those with controls of dimension one, in which case considerable simplifications result with regard to Condition 2. To investigate this somewhat further, let us first note that when the control inputs are scalar, by appropriate rescaling of the cost functions and by redefining B_1, it is always possible to take $R_{11} = R_{12} = R_{22} = 1$. Then, the only weighting term for the controls, left as a variable, is R_{21} which we denote, in this case, by d. Under this set-up, Condition 2 (ii) becomes equivalent to existence of a vector $P' = (p_1,\ldots,p_n)$ so that the solution Z of

$$ZF + F'Z + PB_1'Z + (1-d)\,[P-SB_1]\,B_1'S + Q_2 - Q_1 = 0 \tag{5.8a}$$

satisfies

$$B_2'Z = 0. \tag{5.8b}$$

Investigation of case $n = 1$ leads to the conclusion that either a p^* exists such that $J_1^* = \tilde{J}_1$ $(d \neq 1)$, or that an -Stackelberg strategy for the leader exist such that $J_1^* \leq \tilde{J}_1 + \varepsilon (d = 1)$.

Investigation of the case $n = 2$ leads to the conclusion that existence of a CLS solution for some 2-dimensional examples seems to be the rule rather than an exception. In some examples with $R_{11} = R_{12} = R_{22} = 1, R_{21} \neq 1$, we have failed in all our attempts for finding specific problems which would lead to nonsatisfaction of our sufficiency conditions. For more details the reader is referred to (3).

6. A MORE GENERAL REPRESENTATION

Heretofore, we have restricted the leader's strategies to representations of the functional form (2.13), and have obtained sufficient conditions under which he can enforce his team solution on the follower. It is possible, however, to provide more flexibility for the leader by allowing him a representation of his team strategy more general than (2.13). In particular, if we replace (2.13) with

$$\gamma_1(t,x(s),\ s \leq t) = -R_{11}^{-1}B_1'Sx(t) + P'(t)[x(t) - \bar{x}(t)] + W'(t)[y(t) - \bar{y}(t)], \tag{6.1}$$

where $y(\cdot)$ is an n-dimensional vector function satisfying the differential equation

$$\dot{y} = Cy + Dx\ ,\ y(0) = 0 \tag{6.2}$$

and $\bar{y}(\cdot)$ denotes the solution of (6.2) with $x(\cdot)$ replaced by the optimal team trajectory $\bar{x}(\cdot)$, then the leader has the freedom of choosing the matrix functions $W(\cdot)$, $C(\cdot)$, $D(\cdot)$, in addition to $P(\cdot)$, which clearly provides him a greater degree of flexibility.

A condition similar to Condition 1 and a theorem, similar to Theorem 1 can be derived for this more general set-up. See (3) for explicit formulations. For the scalar case $(n = 1)$ it turns out that this more general class of representations does not possess any superiority on the class of strategies of the form (2.13). It is conjectured that this holds true for $n > 1$ also, since we have failed in our attempts of producing a counterexample to that dictum.

---

*Partially supported by N00014-77-C-0533. Harvard University, Cambridge, MA.

REFERENCES

(1) T. Basar and H. Selbuz, "CLS strategies with applications in the optimal control of multilevel systems", IEEE Transactions on Automatic Control, Vol. AC-24, No. 2, 1979

(2) G.P. Papavassilopoulos and J.B. Cruz, Jr. "Stackelberg and Nash strategies with memory", submitted for publication, 1978.

(3) T. Basar and G.J. Olsder, "Team-Optimal Closed-Loop Stackelberg Strategies in Hierarchical Control Problems. Internal report no.242, Dept. Applied Math., Twente Univ. of Technology, Feb. 1979.

DISTURBANCE ISOLATION VIA HIGH GAIN FEEDBACK*

HASSAN K. KHALIL
Department of Electrical Engineering
and Systems Science
Michigan State University
East Lansing, MI 48824

ABSTRACT

In this paper we discuss the use of high gain state feedback to reduce the effect of disturbances on the outputs of a linear multivariable system. Singular perturbation methods are employed to analyze the high gain feedback control system and derive rules for choosing the feedback strategy to achieve the desired degree of isolation between outputs and disturbances.

INTRODUCTION

In all realistic systems disturbances are included which usually have unfavorable effect on the control system performance. In [1] Wonham defined the disturbance decoupling problem (DDP) via state feedback as follows. Given the system

$$\dot{x} = Ax + Bu + Ev \qquad (1)$$

$$y = Cx \qquad (2)$$

where $x \in R^n$, $u \in R^r$, $y \in R^m$ and $v \in R^p$ are state, input, output and disturbance vectors, respectively, find a state feedback control law

$$u = Fx \qquad (3)$$

such that

$$\text{(I)} \qquad \text{Re}\lambda(A+BF)<0. \qquad (4)$$

$$\text{(II)} \qquad W(t) = C \exp(At + BFt)\, E \equiv 0 \qquad (5)$$

Requirement (II) means that the disturbance v has no effect on the output y. This of course leads to a very restrictive condition on the system parameters. In [2] Willems relaxed (II) by requiring the effect of disturbances on the outputs to be reduced as much as desired rather than being completely eliminated. The degree of coupling between the disturbances and the outputs is defined as follows. Let $y_3(t)$ be the output due to the disturbance and W(t) be the impulse response matrix of the closed loop system from disturbances to outputs, as defined in (5). Then

$$\int_0^\infty y_3^T(t) y_3(t) dt = \int_0^\infty v^T(t)\, W^T(t)\, W(t)\, v(t)\, dt \qquad (6)$$

The coupling between outputs and disturbances is characterized by the index

$$\phi = ||\Phi||_\infty \text{with}$$

$$\Phi = \int_0 W^T(t)\, W(t) dt = E^T SE \qquad (7)$$

where S satisfies the Lyapunov equation

$$S(A+BF) + (A+BF)^T S + C^T C = 0. \qquad (8)$$

---

*This work was supported in part by the National Science Foundation Under Grant No. ENG-7912152.

A precise statement of the disturbance isolation problem (DIP) via state feedback is: given the system (1) and (2) and a positive ε find a state feedback control law (3) such that the closed-loop system is asymptotically stable, requirement I, and

$$\text{(II')} \qquad \phi < \varepsilon \tag{9}$$

The DIP is solvable if for all given positive ε, however small, there exists a feedback matrix F such that (I) and (II') hold. Willems solves the DIP problem by means of the high gain state feedback

$$u = -\frac{1}{\rho} B^T P x \tag{10}$$

where $\rho > 0$ is small enough to have $\|E^T P E\| < \varepsilon$ and where P is the stabilizing solution of the algebraic Riccati equation.

$$PA + A^T P - \frac{1}{\rho^2} PBB^T P + C^T C = 0. \tag{11}$$

The existence of P is guaranteed by requiring (A,B,C) to be stabilizable-detectable. The DIP problem is solvable if and only if $P_o E = 0$ where $P_o = \lim_{\rho \to 0} P$ [2]. Willems approach has two drawbacks. First, it does not give insight into the effect of using high gain feedback because the limiting behavior of the stabilizing solution of the Riccati equation (11) as $\rho \to 0$ is not well understood except in very special cases [3,4]. Second, the use of the control law (10) to acheive the required degree of coupling for very small ε will require solving (11) for small ρ which is numerically ill-conditioned.

In this paper we investigate the use of the high gain state feedback

$$u = \frac{1}{\rho} Fx \tag{12}$$

to solve the DIP problem. We employ the equivalence between singularly perturbed systems and high gain feedback systems [3] to get insight into the effect of using (12). It should be clear that (12) is less general than the control law (10) used by Willems since F in (12) is not function of the small parameter ρ, while in (10) P is a function of ρ. In general, the class of systems for which the DIP problem is solvable using (12) is a subclass of the class of systems for which the DIP problem is solvable. However, the use of the relatively advanced singular perturbation techniques will provide an understanding of the effect of using (12) that is not possible with (10).

Derivation of the main result

We consider the system (1) and (2) and assume that (A,B) is stabilizable and, without loss of generality, that rank (B) =r. Singular perturbations are used to analyze the closed-loop system when the high gain state feedback

$$u = \frac{1}{\mu} Fx \tag{13}$$

is used, where $\mu > 0$ is a small parameter. We start by using a state transformation suggested in [3], namely,

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = Mx \tag{14}$$

where M is a product of elementary row transformations on B such that

$$MB = \begin{pmatrix} 0 \\ B_2 \end{pmatrix} \tag{15}$$

where the rxr matrix B_2 is nonsignular. The transformed system is

$$\dot{x}_1 = A_{11}x_1 + A_{12}x_2 + E_1 v \tag{16}$$

$$\dot{x}_2 = A_{21}x_1 + A_{22}x_2 + B_2 u + E_2 v \tag{17}$$

$$y = C_1 x_1 + C_2 x_2. \tag{18}$$

We are going to use the property that (A_{11},A_{12}) is stabilizable iff (A,B) is stabilizable [3, Lemma 1]. The control law (13) is correspondingly written as

$$u = \frac{1}{\mu}(F_1 x_1 + F_2 x_2). \tag{19}$$

The application of (19) to (16) - (18) results in the singularly perturbed system

$$\dot{x}_1 = A_{11}x_1 + A_{12}x_2 + E_1 v \tag{20}$$

$$\mu\,\dot{x}_2 = (\mu A_{21} + B_2 F_1)\,x_1 + (\mu A_{22} + B_2 F_2)x_2 + \mu E_2 v \tag{21}$$

$$y = C_1 x_1 + C_2 x_2. \tag{22}$$

As usual in singular perturbations we associate with (20) - (22) a reduced system obtained by setting $\mu=0$. Assuming that F_2 is nonsingular the reduced system is

$$\dot{x}_o = (A_{11}-A_{12}F_o)x_o + E_1 v \tag{23}$$

$$y_o = (C_1 - C_2 F_o)x_o \tag{24}$$

where $F_o = F_2^{-1}F_1$. The coupling between the disturbances and the outputs in the reduced system is characterized by $\phi_o = \|\Phi_o\|$ with

$$\Phi_o = \int_0^\infty E_1^T \exp(A_{11}t - A_{12}F_o t)^T (C_1 - C_2F_o)^T (C_1 - C_2F_o)\exp(A_{11}t - A_{12}F_o t)E_1 dt \tag{25}$$

$$= E_1^T S_o E_1 \tag{26}$$

where S_o satisfies the Lyapunov equation

$$S_o(A_{11}-A_{12}F_o)+(A_{11}-A_{12}F_o)^T S_o + (C_1-C_2F_o)^T(C_1-C_2F_o) = 0. \tag{27}$$

Proposition 1: If (A,B) is stabilizable and F_1 and F_2 are chosen to satisfy

$$\mathrm{Re}\,\lambda(A_{11}-A_{12}F_o)<0 \tag{28}$$

and

$$\mathrm{Re}\,\lambda\,(B_2F_2)<0 \tag{29}$$

then there exists $\mu^*>0$ such that for all $\mu \in (0,\mu^*]$ the closed-loop system is asymptotically stable and

$$\phi \leq \phi_o + 0(\mu) \tag{30}$$

Before we prove the proposition we notice that there always exists F_o satisfying (28) since the stabilizability of (A,B) is equivalent to the stabilizability of (A_{11},A_{12}). We also notice that any matrix F_2 satisfying (29) must be nonsingular because B_2 is nonsingular. Hence the reduced system (23),(24) is well-defined.

Proof: The asymptotic satbility of the closed-loop system follows from a well-known singular perturbation theorem [5]. We need only to prove (30). We investigate the asymptotic behavior of the coupling index of the full system as defined in (7). As usual in Lyapunov equations for singularly perturbed system we seek S in the form

$$S = \begin{pmatrix} S_{11}(\mu) & \mu S_{12}(\mu) \\ \mu S_{12}^T(\mu) & \mu S_{22}(\mu) \end{pmatrix}. \tag{31}$$

The Lyapunov equation for S is partitioned into three equations

$$S_{11}A_{11}+A_{11}^T S_{11}+S_{12}(\mu A_{21}+B_2F_1)+(\mu A_{21}+B_2F_1)^T S_{12}^T+C_1^T C_1=0 \tag{32}$$

$$S_{11}A_{12}+S_{12}(\mu A_{22}+B_2F_2)+\mu A_{11}^T S_{12}+(\mu A_{21}+B_2F_1)^T S_{22}+C_1^T C_2=0 \tag{33}$$

$$\mu S_{12}^T A_{12}+\mu A_{12}^T S_{12}+S_{22}(\mu A_{22}+B_2F_2)+(\mu A_{22}+B_2F_2)^T S_{22}+C_2^T C_2=0. \tag{34}$$

Setting $\mu=0$ in (32) - (34) we get

$$S_{11}(0)A_{11}+A_{11}^T S_{11}(0)+S_{12}(0)B_2F_1+F_1^T B_2^T S_{12}^T(0)+C_1^T C_2=0 \tag{35}$$

$$S_{11}(0)A_{12}+S_{12}(0)B_2F_2+F_1^T B_2^T S_{22}(0)+C_1^T C_2=0 \tag{36}$$

$$S_{22}(0)B_2F_2+F_2^T B_2^T S_{22}(0)+C_2^T C_2=0 \tag{37}$$

From (29) it follows that (37) has a unique solution $S_{22}(0)\geq 0$. From (36) we have

$$S_{12}(0)=-S_{11}(0)A_{12}F_2^{-1}B_2^{-1}F_1^T B_2^T S_{22}(0)F_2^{-1}B_2^{-1}-C_1^T C_2F_2^{-1}B_2^{-1}. \tag{38}$$

Substituting (38) into (35) and using (37) to eliminate $S_{22}(0)$, it can be shown that $S_{11}(0)$ satisfies equation (27) replacing S_o, and, by (28), it follows that

$$S_{11}(0) = S_o \tag{39}$$

Using the implicit function theorem yields that there exists $\mu^*>0$ such that for all $\mu<\mu^*$ the solution of (32) - (34) exists and satisfies

$$S_{ij}(\mu) = S_{ij}(0)+0(\mu). \tag{40}$$

Now

$$\Phi = (E_1^T,E_2^T)S\begin{pmatrix} E_1 \\ E_2 \end{pmatrix} = E_1^T S_{11}E_1+2\ E_1^T S_{12}E_2+\ E_2^T S_{22}E_2$$

$$= E_1^T S_o E_1+0(\mu)=\Phi_o+0(\mu) \tag{41}$$

Taking the norms of both sides of (41) proves (30).

Our main result is

Theorem. The DIP problem for the full system (16)-(19) is solvable if the DDP problem for the reduced system (23)-(24) is solvable.

Proof: Recall that the DDP problem for the reduced system is solvable if there exists a matrix F_o such that

$$\mathrm{Re}\,\lambda(A_{11}-A_{12}F_o)<0$$

and

$$(C_1-C_2F_o)\exp(A_{11}t-A_{12}F_o t)E_1\equiv 0 \tag{42}$$

resulting in $\phi_o=0$. Recall also that it is always possible to find F_2 such that $\mathrm{Re}\lambda(B_2F_2)<0$ because B_2 is nonsingular. Construct the state feedback law

$$u=\frac{1}{\mu}(F_2F_o x_1+F_2x_2). \tag{43}$$

Proposition 1 shows that the application of (43) to the full system (16)-(18) results in $\phi=0(\mu)$. Hence ϕ can be made arbitrarily small by choosing μ sufficiently small.

Remark 1: A class of full systems for which the DIP problem is solvable can be determined by characterizing the class of reduced systems for which the DDP problem is solvable. A geometric necessary and sufficient condition for existence of solution of the latter problem is given in the appendix.

Remark 2: When the DDP problem for the reduced system is solvable, the proof of the theorem provides a procedure to construct the feedback law by solving two separate problems to obtain F_o and F_2 and then combining them in the composite control (43). We, therefore, avoid solving ill-conditioned Riccati equation as in Willem's work [2]. The only remaining task to implement (43) is the choice of sufficiently small μ such that $\phi<\varepsilon$ for the given ε. This choice requires computing S for small values of μ which can be achieved by seeking a power series solution of equations (32)-(34).

Remark 3: There are special cases where the solvability of the DDP problem for the reduced system readily follows from the system structure without the need to go through the procedure outlined in the appendix. Two such cases, which are of practical significance, are given here.

Case 1:

$$R_a(E)\subset R_a(B)$$

where $R_a(.)$ denotes the range space. In this case $E_1=0$ which means that the reduced problem is disturbance free. Therefore, any stabilizing matrix F_o will work.

Case 2: $m=r$, rank $(CB)=r$, and the transmission zeros of the system lie in the left half plane. In this case $\mathrm{Re}\,\lambda(A_{11}-A_{12}C_2^{-1}C_1)<0$[3], and $F_o=C_2^{-1}C_1$ solves the DDP problem for the reduced system.

The case $R_a(E)\subset R_a(B)$ is of particular interest because:

First: If the output of the system is taken to be the entire state vector, i.e., C is the identity matrix, then $R_a(E)\subset R_a(B)$ is a necessary and sufficient condition for the solvability of the DIP problem [6].

Second: The feedback control law obtained using the procedure suggested above will be robust to a large class of system parameter uncertainties. This property is stated in Proposition 2.

Proposition 2: Consider the system (1) and (2) with (A,B) being stabilizable and $R_a(E)\subset R_a(B)$. Let F be the feedback matrix in (13) that is obtained using the procedure of the above theorem. Let $\Delta A, \Delta B, \Delta E$ and ΔC be changes in the system parameters such that

(i) $R_a(\Delta B)\subset R_a(B)$, and Rank $(B+\Delta B)=r$,

(ii) $R_a(\Delta E)\subset R_a(B)$, and

(iii) The asymptotic stability of the closed-loop system is preserved, i.e., $\mathrm{Re}\,[A+\Delta A+\frac{1}{\mu}(B+\Delta B)F]<0$ for μ $(0,\mu^*]$ for some $\mu^*>0$.

Then, the application of $u=\frac{1}{\mu}Fx$ to the perturbed system results in a coupling index $\phi=0(\mu)$.

Proof: The closed-loop system is

$$\dot{x} = [A+\Delta A + \frac{1}{\mu}(B+\Delta B)F]x+(E+\Delta E)v \tag{44}$$

$$y = (C+\Delta C)x \tag{45}$$

Using the state transformation (14) the system (44),(45) is transformed into

$$\dot{x}_1=(A_{11}+\Delta A_{11})x_1+(A_{12}+\Delta A_{12})x_2 \tag{46}$$

$$\mu\dot{x}_2 = [\mu A_{21}+\mu\Delta A_{21}+(B_2+\Delta B_2)F_1]x_1+[\mu A_{22}+\mu\Delta A_{22}+(B_2+\Delta B_2)F_2]x_2 + \mu(E_2+\Delta E_2)v \tag{47}$$

$$y = (C_1+\Delta C_1)x_1+(C_2+\Delta C_2)x_2 \tag{48}$$

Clearly the reduced system obtained by setting $\mu=0$ is disturbance free, and since the closed-loop stability is preserved, the arguement of the proof of Proposition 1 can be repeated to show that $\phi=0(\mu)$.

Conclusion

The DIP problem for the full system has been replaced by a DDP problem for a reduced-order system by using a high gain state feedback law. When the DDP problem is solvable, a two step algorithm has been given to choose the feedback matrix. The algorithm is numerically well-conditioned and can be used no matter how high the gain is.

References

1. W.M.Wonham, " Linear Multivariable Control; a Geometric Approach," Springer-Verlag, 1974.
2. J.L. Willems, "Disturbance isolation in linear feedback systems," Int. J. SYSTEMS SCI., Vol. 6, pp. 233-238, 1975.
3. K.K.D. Young, P.V. Kokotovic and V.I.Utkin," A singular perturbation analysis of high-gain feedback systems." IEEE Trans. on Automat. Contr., Vol, AC-22, pp 931-938, 1977.
4. A. Jameson and R.E. O'Malley, Jr., "Cheap control of the time-invariant regulator," Appl. Mth. Opt., Vol.1, pp. 211-219, 1972.
5. R.R. Wilde and P.V.Kokotovic, "Stability of singularly perturbed systems and networks with parasitics," IEEE Trans. Automat. Contr., Vol AC-17 pp 245-246, 1972.
6. H.K. Khalil, "Necessary and sufficient conditions for disturbance isolation," submitted for publication.

APPENDIX

We consider the DDP problem for the system

$$\dot{x} = A_ox+B_ou+E_ov \tag{A.1}$$

$$y = C_ox+D_ou \tag{A.2}$$

Where $x\in X$, $u\in U$ and $y\in Y$. It is required to find (if possible) $F:X\rightarrow U$ such that

$$<A_o+B_oF/\mathcal{E}_o> \subset \mathcal{N}(C_o+D_oF) \tag{A.3}$$

and

$$\mathrm{Re}\lambda(A_o+B_oF)<0 \tag{A.4}$$

where $\mathcal{E}_o = R_a(E_o)$, $\mathcal{N}(.)$ denotes the Null space, and

$$< A_o + B_oF/\mathcal{E}_o > = \mathcal{E}_o + (A_o+B_oF)\mathcal{E}_o + \ldots\ldots + (A_o+B_oF)^{n-1}\mathcal{E}_o.$$

The difference between this problem and the one considered in [1] is the presence of an input term in the output equation. Necessary and sufficient conditions for the solvability of the problem will be derived. We will use the following proposition.

Proposition A.1: Let (A_o,B_o) be stabilizable, $A_oV \subset V \subset X$ and write $\tilde{X} = X/V$ (the factor space of X modulo V). Let $(\tilde{A}, \tilde{B})$ be the pair induced in $\tilde{X}$ by (A_o,B_o). Then $(\tilde{A},\tilde{B})$ is stabilizable.

Proof: Let $R = <A_o/\mathcal{B}_o> \subset X$ with $\mathcal{B}_o = R_a(B_o)$, and X^+ and X^- be the subspaces of the unstable modes and the stable modes of A, respectively. Then

$$(A_o,B_o) \text{ is stabilizable} \Leftrightarrow X^+ \subset R. \qquad \text{(A.5)}$$

We have

$$X = X^+ \oplus X^-$$

and

$$V = V^+ \oplus V^-$$

where $\quad V^+ = V \cap X^+$ and $V^- = V \cap X^-$.

It follows that

$$\tilde{X} = \tilde{X}^+ \oplus \tilde{X}^-$$

where

$$\tilde{X}^+ = X^+/V^+ \text{ and } \tilde{X}^- = X^-/V^-$$

It can be shown that $\tilde{X}^+$ is the subspace of the unstable modes of $\tilde{A}$. Let $P: X \to \tilde{X}$ be the canonical projection. Then

$$\tilde{A}P = PA_o, \tilde{B} = PB_o, \tilde{X}^+ = PX^+ \text{ and}$$

$$PR = P< A_o/\mathcal{B}_o> = P(\mathcal{B}_o + A_o\mathcal{B}_o + \ldots + A_o^{n-1}\mathcal{B}_o)$$

$$= (P\mathcal{B}_o + \tilde{A}P\mathcal{B}_o + \ldots + \tilde{A}^{n-1}P\mathcal{B}_o) = <\tilde{A}/\tilde{\mathcal{B}}>.$$

Finally

$$X^+ \subset R \Rightarrow PX^+ \subset PR \Leftrightarrow \tilde{X}^+ \subset <\tilde{A}/\tilde{B}> \Leftrightarrow (\tilde{A},\tilde{B}) \text{ is stabilizable.} \qquad \text{(A.6)}$$

Using (A.5) and (A.6) completes the proof of Proposition A.1.

Consider the family of subspaces

$$\underline{V} = \{V: V \subset X \ni \exists\, F: X \to U,\ \mathrm{Re}\lambda[(A_o+B_oF)/V] < 0,$$
$$(A_o+B_oF)V \subset V \subset \mathcal{N}(C_o+D_oF)\}.$$

When (A_o,B_o) is stabilizable $\underline{V}$ is nonempty, since $o \in V$, and it is closed under addition; hence, it has a supremal element. Let V^* be the supremal subspace of $\underline{V}$. The solvability of the DDP problem is characterized in the following proposition.

Proposition A.2: The DDP problem for (A.1) and (A.2) is solvable, i.e., there exists $F: X \to U$ satisfying (A.3) and (A.4), if and only if (A_o,B_o) is stabilizable and $V^* \supset \mathcal{E}_o$.

Proof: It is sufficient to show that when (A_o, B_o) is stabilizable the problem is solvable iff $V^* \supset \mathcal{E}_o$. Thus we assume that (A_o,B_o) is stabilizable. Let $V^* \supset \mathcal{E}_o$. There exists F_o such that

$$\mathrm{Re}\lambda\,[(A_o+B_oF_o)/V^*] < 0 \qquad \text{(A.7)}$$

and

$$\langle A_o+B_oF_o/V^*\rangle \subset \mathcal{N}(C_o+D_oF_o). \qquad \text{(A.8)}$$

Since $(A_o+B_oF_o,\ B_o)$ is stabilizable, by Proposition A.1 the corresponding pair $(\overline{A},\overline{B})$ induced in X/V^* is also stabilizable. Hence, there exists $\overline{F}:X/V^*\to U$ such that

$$\mathrm{Re}\lambda\,(\overline{A}+\overline{B}\overline{F})<0. \qquad \text{(A.9)}$$

Let $Q : X\to X/V^*$ be the canonical projection and set $F=F_o+\overline{F}Q$.

Requirement (A.4) follows from (A.7) and (A.9). Requirement (A.3) is satisfied since $\langle A_o+B_oF/\mathcal{E}_o\rangle \subset \langle A_o+B_oF/V^*\rangle = \langle A_o+B_oF_o/V^*\rangle \subset \mathcal{N}(C_o+D_oF_o)$.

Alternatively, suppose that there exists F satisfying (A.3) and (A.4). Take $V=\langle A_o+B_oF/\mathcal{E}_o\rangle$. Clearly $\mathcal{E}_o\subset V$ and $V\in \underline{V}$ implying that $\mathcal{E}_o\subset V^*$.

To verify the condition $V^*\supset\mathcal{E}_o$, we need an algorithm to find V*. To this end let us take the direct sum $X\oplus Y$ and define the maps

$$A_e \triangleq \begin{pmatrix} A_o & 0 \\ C_o & 0 \end{pmatrix} : X\oplus Y\to X\oplus Y, \qquad B_e \triangleq \begin{pmatrix} B_o \\ D_o \end{pmatrix} = U\to X\oplus Y.$$

Consider the family of subspaces

$$\underline{Ve} = \{\, V:\ V\subset X\curlyvee\exists F_e,\ \mathrm{Re}\lambda[(A_e+B_eF_e)/V] < 0,$$
$$(A_e+B_eF_e)V\subset V \,\}.$$

It can be easily verified that

$$\underline{V} = \underline{V}_e \qquad \text{(A.10)}$$

An algorithm to find the supermal element of $\underline{V}_e$ is given in [1,pp.117-118] .

DESIGN OF OPTIMAL TRACKING DISCRETE CONTROL SYSTEM[+]

Y. BAR-NESS[++]
Lefschetz Center for Dynamical Systems
Division of Applied Mathematics
Brown University
Providence, Rhode Island 02912

J. FEINSTEIN
School of Engineering
Tel Aviv University
ISRAEL

ABSTRACT

Output feedback with a series compensator is considered for controlling SISO discrete systems. MSE criterion of design is used in tracking a given input sequence. The plant can be unstable and/or non-minimum phase, while the compensator is constrained to be physically realizable and closed-loop configuration is both stable and physically realizable. Using variational methods the required compensator is found, and shown to be the necessary and sufficient optimal solution.

I. INTRODUCTION

The design of a discrete system to track a given input sequence is an important problem. The use of the minimum squared error (MSE) as a criterion of performance in such problems is well accepted. Kue[1] in treating this problem, did not take into consideration either the system configuration or the plant properties, so that in his solution the closed-loop may be unstable and the compensator obtained might not be physically realizable.

In this paper, output feedback together with a series compensator is considered for controlling the system to obtain the desired objective. This procedure is particularly suitable if the required compensator is constrained to be physically realizable(*) and the resultant closed-loop system is to be both stable and physically realizable.

Consider the physically realizable plant transfer function

$$G(z) = \frac{p^{+}(z)p^{-}(z)}{q^{+}(z)q^{-}(z)} \qquad (1)$$

where p^{+} and q^{+} are polynomials whose zeros are located inside the unit circle and p^{-} and q^{-} are polynomials whose zeros are located outside this circle.

[+] This research was supported by the Air Force Office of Scientific Research under AF-AFOSR 77-3063.

[++] Author on sabbatical leave from the School of Engineering, Tel Aviv University, Tel Aviv, Israel. Since Sept.1979 he has been at Systems Eng. Dept.& Valley Forge Research Center, Moore School,U.of Pen.Phila.Pa.19104

(*) System is physically realizable iff the difference between the degrees of its transfer function's denominator and nominator, $\ell_{(\cdot)}$, is not less than zero.

For the given input sequence U_k and its corresponding output sequence W_k the squared error is given by,

$$J = \sum_{k=0}^{\infty} e_k^2 = \sum_{k=0}^{\infty} (U_k - W_k)^2. \tag{2}$$

Using Parseval's theorem, we get

$$J = \frac{1}{2\pi j} \oint E(z)E(z^{-1})\frac{dz}{z} \tag{3}$$

where

$$E(z) = U(z) - W(z) = \sum_{k=0}^{\infty} e_k z^{-k} \tag{4}$$

is the z transform of the error sequence.

Denoting the closed-loop transfer function by $T(z)$, we have

$$E(z) = (1-T(z))U(z) \tag{5}$$

with

$$T(z) = G(z)D(z)/[1+G(z)D(z)] \tag{6}$$

where $D(z)$ is the compensator transfer function that has to be obtained such that the squared error is minimized.

To solve the minimization problem, it is possible to use the Wiener filter approach and find $T(z)$. $T(z)$ may even be required to be a physically realizable solution. Nevertheless, with $T(z)$ available, the required compensator is obtained from equation (6)

$$D(z) = 1/G(z)[T(z)/1-T(z)]. \tag{7}$$

Clearly, $D(z)$ must cancel some of the zeros or poles of the plant $G(z)$, which might be outside the unit circle, thus yielding an unstable closed-loop system. To cope with the problem of zero cancellation in the continuous case, Newton[2] and, later, Bongiorno[3] and Weston[4] restricted the plant to be asymptotically stable. Instead, as is done in this paper, it is reasonable to require that $D(z)$ is allowed to cancel only those poles and zeros of $G(z)$ which are inside the unit circle. That is

$$D(z) = q^+(z)\alpha(z)/p^+(z)\beta(z) \tag{8}$$

where $\alpha(z)$ and $\beta(z)$ are polynomials, yet to be determined.

To obtain these polynomials we define an appropriate variational problem on a space of polynomials such that it takes into account the requirements that $D(z)$ be physically realizable and that $T(z)$ is both stable and physically realizable. From the solution of the variation problem we obtain the necessary and sufficient condition which $D(z)$ - in particular $\alpha(z)$ and $\beta(z)$ - has to satisfy. It turns out that for $D(z)$ to be the required optimal solution, $\alpha(z)$ and $\beta(z)$ must satisfy some polynomial equation whose parameters are easily obtained from the input transform $U(z)$ and the plant $G(z)$ - see Theorem 4.6. Furthermore, we show that under a certain polynomial degree condition, the solution of this polynomial equation, $\alpha(z)$ and $\beta(z)$, is unique - see Lemma 4.4. With this, $D(z)$ is the required optimal compensator - see Theorem 4.7. Finally, we calculate the value of the optimal performance for comparison purpose. Two examples supplement the theoretical results.

Although we consider the case of a given input sequence, nevertheless, it is possible to extend this work to the case when the input includes noise or disturbances with a given spectral density.

Youla et al[5] treated a related problem for the continuous case, and (as in this work) the plant was allowed to be unstable and/or nominimum phase. In addition to the fundamental difference between the discrete and continuous systems, Youla's work differs from ours in several respects. The approaches are quite different. Youla's paper uses spectral factorization and some of the implicit assumption, might be rather restrictive (e.g. assumption 6). Secondly the resultant controller might be physically unrealizable depending on the given input or plant, a situation which might also be considered restrictive, particularly if flexible implimentation of this controller is required (see [5] footnote on page 7). Finally, Theorem 1 of Youla constitutes only a necessary condition. Another reference which may be mentioned is Volgin[6], who discussed the problem without adequately solving it.

II. STATING THE PROBLEM

Consider the discrete control system with output feedback and a series compensator. Let the plant $G(z)$ be given by equation (1). For a given input sequence U_k, it is required to find the polynomials $\alpha(z)$ and $\beta(z)$ of the compensator $D(z)$ (equation (8)), such that:

(1) $D(z)$ is physically realizable.

(2) The closed-loop transfer function $T(z)$ (equation (6)) is stable and physically realizable.

(3) The squared error between the given input sequence U_k and its corresponding output sequence W_k is minimized. (Equivalently, the cost functional of equation (3) is minimized.)

III. ON THE CONSTRAINTS OF THE VARIATIONAL PROBLEM

First, we consider some results related to the constraints of the variational problem. In terms of the polynomials $\alpha(z)$ and $\beta(z)$, the closed-loop transfer function is given by

$$T(z) = p^{-}(z)\alpha(z)/v(z) \tag{9}$$

where,

$$v(z) = p^{-}(z)\alpha(z) + q^{-}(z)\beta(z). \tag{10}$$

Because of lack of space we will sometimes state our claims without proving them.

Proposition 3.1: For $T(z)$ to be physically realizable it is necessary and sufficient that the degree δ_v of the polynomial $v(z)$ in equation (10) satisfies

$$\delta_v = \max(\delta_{p^-\alpha}, \delta_{q^-\beta}). \quad \blacksquare \tag{11}$$

Proposition 3.2. Assume that the plant $G(z)$ is physically realizable, then

(a) For both the closed-loop transfer function $T(z)$ and the compensator $D(s)$ to be physically realizable it is necessary that

$$\delta_v = \delta_{q^-\beta^-} > \delta_{q^-\alpha}, \tag{12}$$

in which case,

$$\ell_T = \ell_G + \ell_D \tag{13}$$

where $\ell_{(\cdot)}$ is the excess denominator-nominator zeros.

(b) On the other hand, if

$$\delta_v = \delta_{q^-\beta^-} > \delta_{p^-\alpha} \tag{14}$$

and

$$\ell_T \geq \ell_G. \tag{15}$$

Then both $T(z)$ and $D(z)$ are physically realizable. ▮

IV. THE SOLUTION OF THE VARIATIONAL PROBLEM

For any given polynomial $x(z)$ of degree δ_x, let

$$\tilde{x}(z) \triangleq z^{\delta_x} x(z^{-1}). \tag{16}$$

If $x(z) = \sum_{i=0}^{N} a_i z^i$, $a_n \neq 0$ $\delta_x = n$, then clearly $\tilde{x}(z) = z^n \sum_{i=0}^{N} a_i z^{-i} = \sum_{i=0}^{n} a_i z^{n-i}$. Also, if $x(z)$ has no zeros at $z = 0$ (i.e., $a_0 \neq 0$), then

$$\delta_{\tilde{x}} = \delta_x. \tag{17}$$

Proposition 4.1: Let $\Phi(z)$ be a rational function of z whose zeros and poles are symmetrically located with respect to the unity circle, i.e.,

$$\Phi(z) = U(z)U(z^{-1}).$$

Then, it is possible to factor $\Phi(z)$

$$\Phi(z) = \tau^+(z)\tau^-(z)/\rho^+(z)\rho^-(z) \tag{18}$$

where the zeros of τ^+ and ρ^+ are located inside the unit circle and those of τ^- and ρ^- are outside it, such that

$$\tilde{\tau}^+(z) = \tau^-(z) \tag{19}$$

$$\tilde{\rho}^+(z) = \rho^-(z) \tag{20}$$

and

$$\delta_{\tau^+} = \delta_{\rho^+}. \tag{21}$$

We are now in a position to set the proper variational problem. Substituting equation (5) into equation (3) yields

$$J = \frac{1}{2\pi j} \oint (1-T(z))(1-T(z^{-1}))\Phi(z)\frac{dz}{z} \tag{22}$$

where $T(z)$ is given by equations (9) and (10) and $\Phi(z)$ by equation (18). Therefore,

$$J = \frac{1}{2\pi j} \oint \frac{q^{-}(z)\beta(z)}{v(z)} \frac{q^{-}(z^{-1})\beta(z^{-1})}{v(z^{-1})} \frac{\tau^{+}(z)\tau^{-}(z)}{\rho^{+}(z)\rho^{-}(z)} \frac{dz}{z} .$$

Using equations (16) and (17) we have

$$J = \frac{1}{2\pi j} \oint \frac{q^{-}(z)\beta(z)}{v(z)} \frac{\tilde{q}^{-}(z)\tilde{\beta}(z)}{\tilde{v}(z)} \frac{\tau^{+}(z)\tau^{-}(z)}{\rho^{+}(z)\rho^{-}(z)} \frac{dz}{z} . \tag{23}$$

Hence, J can be considered as a functional on the linear space P of pairs of polynomials $(\alpha(z),\beta(z))$, which is defined in the Appendix. With the norm

$$||(\alpha,\beta)|| = \max\left(\sum_{i=0}^{\delta_\alpha} |\alpha_i|, \sum_{i=0}^{\delta_\beta} |\beta_i| \right) \tag{24}$$

defined on P, it can be shown that J is a differentiable functional on P whose first variation is given by

$$J[h_1,h_2] = \frac{-1}{\pi j} \oint \frac{q^{-}(z)\ (z)}{v(z)} \frac{\tilde{p}^{-}(z)\tilde{q}^{-}(z)}{[v(z)]^2} \frac{\tau^{+}(z)\tau^{-}(z)}{\rho^{+}(z)\rho^{-}(z)} z^{\ell_T}(\tilde{\beta}(z)\tilde{h}_1(z) - \tilde{\alpha}(z)\tilde{h}_2(z))\frac{dz}{z} \tag{25}$$

where h_1 and h_2 are the variation of α and β.

Lemma 4.2: Consider the linear space P, defined in the Appendix with the norm of equation (24). If

$$\delta_\alpha > \delta_{q^-} + \delta_{\tau^+} \tag{26}$$

and

$$\delta_v = \delta_{q^-\beta} \geq \delta_{p^-\alpha} \tag{27}$$

then for $(\alpha,\beta) \varepsilon P$ to be extremum of the cost functional of equation (22), it is necessary that

$$c(z) = \frac{q^{-}(z)\beta(z)\tilde{p}^{-}(z)\tilde{q}^{-}(z)\tau^{+}(z)\tau^{-}(z)z^{\ell_T}}{v(z)[\tilde{v}(z)]^2\rho^{+}(z)\rho^{-}(z)z} \tag{28}$$

where

$$v(z) = p^{-}(z)\alpha(z) + q^{-}(z)\beta(z) \tag{29}$$

has no poles inside the unit circle.

Proof: For J to have an extremum at $(\alpha,\beta) \varepsilon P$, the variation of equation (30) must vanish for every polynomial increment $h_1(z)$ and $h_2(z)$. Define

$$A(z) \underline{\Delta} \tilde{p}^{-}(z)\tilde{q}^{-}(z)\beta(z)\tau^{+}(z)\tau^{-}(z)p^{-}(z)z^{\ell_T}/d(z)$$

and

$$B(z) \triangleq zv(z)\rho^+(z)/d(z)$$

so that $A(z)$ and $B(z)$ have no common zeros (i.e., $d(z)$ is the corresponding g.c.d.). Also, let

$$\mu(z) \triangleq (\tilde{\beta}(z)\tilde{h}_1(z) - \tilde{\alpha}(z)\tilde{h}_2(z)).$$

The plynomials $\tilde{v}(z)$ and $\rho^-(z)$ have no zeros inside the unit circle and can be disregarded. Clearly,

$$\delta_B \leq 1 + \delta_v + \delta_{\rho^+}. \tag{30}$$

For a given $\alpha(z)$ and $\beta(z)$, $h_1(z)$ and $h_2(z)$ can be selected such that $\mu(z)$ could have zero of order $\nu \leq \delta_\alpha + \delta_\beta$ anywhere in the complex plane, and particularly, inside the unit circle. From equations (26), (27), (30) and since $\delta_{\rho^+} = \delta_{\tau^+}$, by equation (20) we deduce that $\delta_\mu \geq \delta_B - 1$. Applying Lemma B-2 our claim follows.

Lemma 4.3: Assume that the closed-loop transfer function $T(z)$ is stable; then for every $\ell_T \geq 0$, for $c(z)$ to be analytic inside the unit circle, it is necessary that

$$v(z) = \tau^+(z)p^-(z)q^-(z)z^{\ell_T} \tag{31}$$

$$\beta(z) = z\rho^+(z)\pi(z) \tag{32}$$

where $\pi(z)$ is some polynomial which is together with $\alpha(z)$ must satisfy the following equation

$$p(z)\alpha(z) + q^-(z)\cdot z\rho^+(z)\pi(z) = \tau^+(z)p^-(z)q^-(z)z^{\ell_T}. \tag{33}$$

If, furthermore,

$$\delta_v = \delta_{q^-\beta} \geq \delta_{p^-\alpha} \tag{34}$$

then it is necessary that

$$\delta_\pi = \delta_{p^-} + \ell_T - 1 \tag{35}$$

and

$$\delta_\alpha = \delta_{\tau^+} + \delta_{q^-} \tag{36}$$

where $\ell_T \triangleq \delta_{q^-\beta} - \delta_{p^-\alpha}$.

On the other hand, if $\alpha(z)$ and $\pi(z)$ are a polynomial pair solution to equation (33), then with $\beta(z)$ as in (32), $c(z)$ is analytic inside the unit circle.

Proof: For $T(z)$ to be stable $v(z)$ must contain only zeros inside the unit circle. Since ℓ_T might be zero then for $c(z)$ to be analytic inside the unit circle we must have

$$v(z)z\rho^+(z)\pi(z) = \beta(z)\tilde{p}^-(z)\tilde{q}^-(z)\tau^+(z)z^{\ell_T} \tag{37}$$

for some polynomial $\pi(z)$. Substituting equation (29), and rearranging terms we have

$$\beta(z)[\tilde{p}^-(z)\tilde{q}^-(z)\tau^+(z)z^{\ell_T} - zq^-(z)\rho^+(z)\pi(z)] = p^-(z)\alpha(z)\pi(z)\rho^+(z)z. \tag{38}$$

Since $\beta(z)$ and $p^{-}(z)\alpha(z)$ are relatively prime and $\pi(z)$ is arbitrary, equation (32) follows. And consequently, from equation (36) $\pi(z)$ must satisfy equation (33). Substituting equation (32) in equation (33) results in equation (31).

Furthermore, using equation (34) we have

$$\delta_{\beta} = \delta_{v^-} - \delta_{q^-}. \tag{39}$$

Since $p^{-}(z)$ and $q^{-}(z)$ have no zeros inside the unit circle, $\delta_{\tilde{p}^-} = \delta_{p^-}$ and $\delta_{\tilde{q}^-} = \delta_{q^-}$, and from equation (31) we have

$$\delta_v = \delta_{\tau^+} + \delta_{p^-} + \delta_{q^-} + \delta_T. \tag{40}$$

Also, from equation (32) we have

$$\delta_{\beta} = 1 + \delta_{\rho^+} + \delta_{\pi}. \tag{41}$$

Combining equations (39), (40) and (41) and using equation (21), equation (35) follows. Also, from equation (9) $\delta_T \underline{\Delta}\, \delta_v - \delta_{p^-\alpha}$, and substituting for δ_v from equation (40), equation (36) follows after simple manipulations.

On the otherhand, if we pick $\pi(z)$ and $\alpha(z)$ to be solution of equation (33) and take $\beta(z)$ as in equation (32), then clearly $v(z)$ will have the form of equation (31) and using equation (28) $c(z)$ is obviously analytic inside the unit circle.

Lemma 4.4: The solution pair of polynomials $\pi(z)$ and $\alpha(z)$ of equation (33) which satisfy equation (35) is unique for all $\ell_T \geq \delta_G$.

Proof: Assume that $\tau^{+}(z)\tilde{p}^{-}(z)\tilde{q}^{-}(z)z^{\ell_T}$ and $z\rho^{+}(z)$ are relatively prime- To solve equation (33) for the unknown polynomials (z) and $\beta(z)$, notice that the number of unknowns (the coefficients of these polynomials) is $n_1 = \delta_{\pi} + \delta_{\alpha} + 2$. Using equations (35) and (36) $n_1 = \delta_{p^-} + \ell_T + \delta_{\tau^+} + \delta_{q^-} + 1$. On the otherhand, the number of scalar equations obtained by equating the coefficients of different powers is given by the degree of the polynomial on the left $n_2 = \delta_{\tau^+} + \delta_{\tilde{p}^-} + \delta_{\tilde{p}^-} + \ell_T + 1$. Since, $\delta_{\tilde{p}^-} = \delta_{p^-}$ and $\delta_{\tilde{q}^-} = \delta_{q^-}$ it follows that $n_1 = n_2$.

Now, the polynomials $p^{-}(z)$ and $q^{-}(z)z\rho^{+}(z)$ are relatively prime and hence their resultant is nonzero, and the matrix of the coefficients is nonsingular. This ensures the existence and uniqueness of the solution.

If the polynomials $\tau^{+}(z)p^{-}(z)q^{-}(z)z^{\ell_T}$ and $z\rho^{+}(z)$ are not relatively prime, we denote their g.c.d. by $d(z)$. Let $\alpha(z) = d(z)\alpha'(z)$, $z\rho^{+}(z) = d(z)\rho'(z)$, $\tau^{+}(z)\tilde{p}^{-}(z)\tilde{q}^{-}(z)z^{\ell_T} = d(z)v'(z)$. Equation (33) thus becomes $p^{-}(z)\alpha'(z) + q'(z)\rho'(z)\pi(z) = v'(z)$.

This leaves the number of equations and unknowns equal and, therefore, the previous argument of existence and uniqueness for the pair $\alpha(z)$ and $\pi(z)$ follows.

Lemma 4.5. If

$$c(z) = \frac{q^-(z)\beta(z)p^-(z)q^-(z)\tau^+(z)\tau^-(z)z^{\ell_T}}{v(z)[\tilde{v}(t)]^2\rho^+(z)\rho^-(z)z} \tag{28}$$

is analytic inside the unit circle, then the second variation of the cost functional of equation (21) is given by

$$\mu^2 J[h_1,h_2] = \frac{1}{\pi j}\oint \frac{\tilde{p}^-(z)\tilde{q}^-(z)}{[v(z)]^2}\,\Phi(z)z^{\ell_T}\,\frac{p^-(z)q^-(z)}{[\tilde{v}(z)]^2} \tag{42}$$
$$\cdot\,[\tilde{\beta}(z)\tilde{h}_1(z) - \tilde{\alpha}(z)\tilde{h}_2(z)][\beta(z)h_1(z) - \alpha(z)h_2(z)]\frac{dz}{z}$$

and is positive semidefinite.

Theorem 4.6 (necessary condition): Let the input sequence for the discrete control system with output feedback be U_k and its z-transform $U(z)$. Define

$$\Phi(z) \underline{\Delta} U(z)U^{-1}(z) = \frac{\rho^+(z)\rho^-(z)}{\tau^+(z)\tau^-(z)}\,. \tag{18}$$

If the plant $G(z)$ is physically realizable and given by equation (1) then for the compensator $D(z)$, given by equation (2) to be the optimal solution to the problem stated in Section II, it is necessary that $\beta(z) = z\rho^+(z)\pi(z)$, $v(z) = \tau^+(z)\tilde{p}^-(z)\tilde{q}^-(z)z^{\ell_T}$ where $v(z) = p^-(z)\alpha(z) + q^-(z)\beta(z)$, and $\pi(z)$ is a solution to the polynomial equation $p^-(z)\alpha(z) + q^-(z)z\rho^+(z)\pi(z) = \tau^+(z)p^-(z)q^-(z)z^{\ell_T}$ with $\delta_\pi = \delta_{p^-} + \ell_T - 1$, ($\ell \underline{\Delta} \delta_{q^-\beta} - \delta_{p^-\alpha}$), $\delta_\alpha = \delta_{\tau^+} + \delta_{q^-}$ and where $\tilde{q}^+, \tilde{q}^-, \tilde{p}^+$ and $\tilde{p}^-$ are defined by equation (16).

Proof: Using Proposition 3.2, we have $\delta_v = \delta_{q^-\beta} > \delta_{p^-\alpha}$ and using Lemma 4.2 and 4.3 our claim follows.

Theorem 4.7 (sufficient condition): For the discrete control system with output feedback and physically realizable plant as in Theorem 4.6, let $\pi(\alpha)$ and $\alpha(z)$ be the solution of

$$p^-(z)\alpha(z) + q^-(z)z\rho^+(z)\pi(z) = \tau^+(z)p^-(z)q^-(z)z^{\ell_T} \tag{33}$$

satisfying $\delta_\pi = \delta_{p^-} + \ell_T - 1$, $\delta_\alpha = \delta_{\tau^+}\ \delta_{q^-}$ where $\ell_T \geq 0$ and the different polynomials are specified in Theorem 3.9.

Let $\beta(z) = z\rho^+(z)\pi(z)$ then using $D(z) = q^+(z)\alpha(z)/p^+(z)\beta(z)$ as a series compensator, the closed-loop system is stable, the first variation will vanish and the second variation is positive semi-definite for all increments in $\alpha(z)$ and $\beta(z)$. If, furthermore, ℓ_T is chosen so that $\ell_T \geq \ell_G$, then $D(z)$ is also physically realizable.

Proof: First using Lemma 4.4, the pair $\alpha(z)$, and $\pi(z)$ is unique. From equation (8) $T(z) = p^-(z)\alpha(z)/v(t)$ where $v(z)$ is $v(z) = \tau^+(z)\tilde{p}^-(z)\tilde{q}^-(z)z^{\ell_T}$ due to the way $\beta(z)$ is selected. Hence, $T(z)$ is stable, and $\delta_v = \delta_{\tau^+} + \delta_{q^-} + \delta_{p^-} + \ell_T$. By Lemma 4.3, $c(z)$ (equation (28) is analytic inside the in unit circle and, by equation (25), the first variation is zero. Also, by Lemma 4.5 the second variation is positive semidefinite for all increments of the polynomials $\alpha(z)$ and $\beta(z)$.

Now, by our choice of $\pi(z), \alpha(z)$ and consequently of $\beta(z)$, $\delta_{q^-\beta} = \delta_{q^-} + 1 + \delta_{\rho+} + (\delta_{p^-} + \ell_T - 1)$, $\delta_{p^-\alpha} = \delta_{p^-} + \delta_{\tau+} + \delta_{q^-}$. Therefore, using $\delta_{\tau+} = \delta_{\rho+}$ and $\ell_T \geq 0$ we have $\delta_v = \delta_{q^-\beta} \geq \delta_{p^-\alpha}$, and if we choose $\ell_T \geq \ell_G$ then, by Proposition 3.2 the compensator $D(z)$ is physically realizable.

V. THE MINIMUM SQUARE ERROR

For the unconstrained problem we could have used the Wiener solution to obtain the open-loop optimal filter. For the case of simple poles, the optimal filter transfer function (our closed-loop transfer function) obtained using partial fraction expansion[1] has no excess of poles over zeros, i.e., $\ell_T = 0$.

Realizing this optimal Wiener filter $T(z)$ using series compensator configuration, we have that $D(z) = 1/G(z) \cdot [T(z)/1-T(z)]$. Hence, if $\ell_G > 0$, $\ell_D = -\ell_G$ and $D(z)$ is physically unrealizable. By increasing ℓ_T, $D(z)$ becomes realizable although, obviously, the cost increases. One possible physically realizable solution can be obtained by choosing the minimal possible ℓ_T for which $\ell_D = 0$; that is $\ell_T = \ell_G$. Therefore, using Theorem 4.7,

$$D(z) = \alpha(z)q^+(z)/zp^+(z)\rho^+(z)\pi(z) \tag{43}$$

where $\alpha(z)$ and $\pi(z)$ are the solution pair of equation (33).

It is possible to prove using some lenghthy manipulation that the minimum squared error is given by

$$J = \sum \text{Res}\left[\frac{\pi(z)\tilde{\pi}(z)}{z^{\ell_G} p^-(z)\tilde{p}^-(z)}\right] \tag{44}$$

where the summation is on all the residues inside the unit circle; i.e. at the poles of $z^{\ell_G} p^-(z)$.

VI. EXAMPLE

(1) Consider the plant $G(z) = z - 2/(z-1{,}2)(z-0.5)$ **and the** transformed step input sequence $U(z) = z/z-1$.. From equation (1) we get, $q^+(z) = z - 0.5$, $q^-(z) = z - 1{,}2$, $p^+(z) = 1$, $p^-(z) = z - 2$ and by equation (18) $\Phi(z) = z/(z-1)(1-z)$. Hence, $\tau^+(z) = z$, $\tau^-(z) = 1$, $\rho^+(z) = z-1$, $\rho^-(z) = 1 - z$. Equation (33) becomes $(z-2)\alpha(z) + (z-1{,}2)z(z-1)\pi(z) = z \cdot (1-2z)(1-1{,}2z) \cdot z$ where we chose $\ell_T = \ell_G = 1$. Dividing both sides by z, we have, in the notation of Lemma 4.4, $(z-2)\alpha'(z) + (z-1{,}2)(z-1)\pi(z) = z(1-2z)(1-1{,}2z)$.

The unique solution of this equation that satisfies $\delta_{\alpha'} = \delta_\alpha - 1 = \delta_{\tau+} + \delta_{q^-} - 1 = 1$ and $\delta_\pi = \delta_{p^-} + \ell_T - 1 = 1$ is given by $\pi(z) = 5.7 + 2.4z$, $\alpha'(z) = 3.42 - 3.62z$. The compensator $D(z)$ (equation (43)) is, therefore, $D(z) = (z-1/2)(3.42-3.62z)/(z-1)(5.7+2.4z)$ and using equation (44), the minimum square error equals 17.31.

VII. CONCLUSIONS

Output feedback with a series compensator were considered, in controlling the SISO discrete system to track a given input sequence. MSE criterion of design was used in a variational problem whose solution yielded the required compensator transfer function $D(z)$. This compensator was constrained to be physically realizable and the resultant closed-loop system to be both stable and physically realizable.

It turns out that for $D(z)$ to be the required optimal solution, it must include two polynomials $\alpha(z)$ and $\beta(z)$ that have to satisfy some polynomial equation whose parameters are obtained from the input transform and the plant. On the other hand using the solution of this polynomial equation (which, under certain polynomial degree conditions is unique) $D(z)$ is the required optimal compensator. An example that was worked out presents the detailed steps required.

Acknowledgment.

The first author would like to thank Professor W. Fleming for helpful discussions and Professor H. Kushner for his comments on the final manuscript.

APPENDIX

Consider the space P of pairs of polynomials $(\alpha(z), \beta(z))$ of degrees δ_α and δ_β respectively, i.e.

$$\alpha(z) = \sum_{i=0}^{\delta_\alpha} \alpha_i z^i \tag{B-1}$$

$$\beta(z) = \sum_{i=0}^{\delta_\beta} \beta_i z^i \tag{B-2}$$

(where α_i and β_i are numbers) such that

$$\delta_v = \delta_{q^-} + \delta_\beta \geq \delta_{p^-} + \delta_\alpha \tag{B-3}$$

where $v(z) = p^-(z)\alpha(z) + q^-(z)\beta(z)$. Clearly, P is a linear space.

Lemma B-1. Let $A(z)$ and $B(z)$ be two given relatively prime polynomials. If

$$\oint \frac{A(z)}{B(z)} \mu(z)dz = 0$$

for any $\mu(z)$ of a given degree δ_μ, $\delta_\mu \geq \delta_B - 1$, then $A(z)/B(z)$ must be analytic inside the contour of integration.

Proof: By contradiction, assume that $B(z)$ has a pole of multiplicity ν, $\nu \leq \delta_B$ inside the contour of integration, i.e., $B(z) = C_1(z)(z-\alpha)^\nu$, $C_1(z) \neq 0$ for some α. Choose $\mu(z)$, to have a pole at α of multiplicity $\nu - 1$, i.e., $\mu(z) = C_2(z)(z-\alpha)^{\nu-1}$, $C_2(\alpha) \neq 0$. Hence, by the residue theorem

$$\oint \frac{A(z)}{B(z)} \mu(z)dz = \frac{A(\alpha)C_2(\alpha)}{C_1(\alpha)} \neq 0$$

which contradicts our assumption.

REFERENCES

[1] Kuo,,B.C., Discrete-Data Control Systems, Englewood Cliffs, New Jersey, Prentice-Hall, 1970.

[2] Newton, G.C., L.A. Gould and J.F. Kaiser, Analytical Design of Linear Feedback Control, New York, Wiley, 1957.

[3] Bongiorno, J.J., "Minimum Sensitivity Design of Linear Multivarible Feedback Control Systems by Matrix Spectral Factorization", IEEE Trans. on Automatic Control, AC-14, 1969, pp. 665-673.

[4] Weston, J.E. and J.J. Bongiorno, "Extension of Analytical Design Techniques to Multivariable Feedback Control Systems", IEEE Trans. on Automatic Control, AC-17, 1972, pp. 613-620.

[5] Youla, P.C., H.A. Jabor and J.J. Bongiorno, "Modern Wiene-Hopf Design of Optimal Controller-Part I: The Single Input, Single Output Case", IEEE Trans. on Automatic Control, AC-21, 1972, pp. 3-13.

[6] Volgi, L.N., Elements of Theory of Computer Control, Sovyetskoye Radio, 1962.

MICROPROCESSOR BASED COMPENSATION FOR NONLINEAR FRICTIONS

DONG H. CHYUNG
Electrical and Computer Engineering
University of Iowa
Iowa City, Iowa 52242

ABSTRACT

A digital compensation scheme, which is based on a microprocessor, for nonlinear frictions such as Coulomb friction and stiction in a linear position control system is presented.

SUMMARY

One of the problems which occur frequently in a mechanical position control system is the non-zero steady state error due to nonlinear frictions such as Coulomb friction and stiction. The steady state error may be reduced by increasing the gain or by gearing down the motor output. It may also be reduced by using the integral of the error as the control input. However, these schemes may also introduce instability, excessive overshoot and long settling time. In this paper, we present another method which is based on a microprocessor to reduce the steady state error.

The system investigated was a d.c. motor driven position control system shown in Fig. 1.

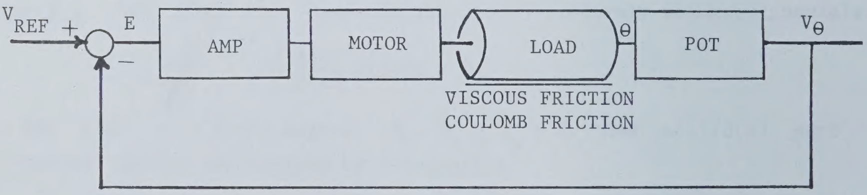


Fig. 1 DC MOTOR POSITION CONTROL SYSTEM

---

*This research was supported in part by NSF Grant No. ENG 78-00814 and U.S. Army Grant No. DAAK10-78-C-0325.

In addition to the Coulomb friction, the system contained some striction as well. However, the compensation scheme was mainly for countering the Coulomb friction. The mathematical model of the system is given in Fig. 2.

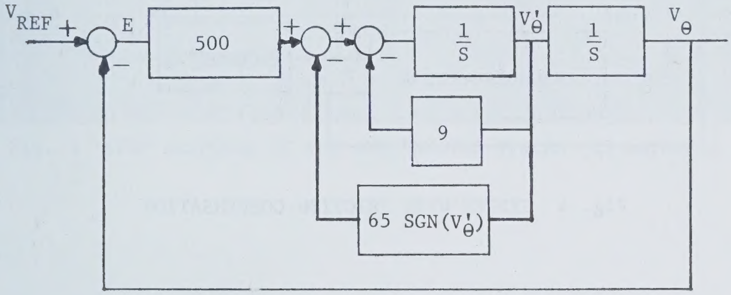

V_{REF}
+
E
−
500
+
+
−
+
−
$\frac{1}{S}$
V'_θ
$\frac{1}{S}$
V_θ
9
65 SGN(V'_θ)

Fig. 2 MATHEMATICAL MODEL

When a step input of 1V was applied, the response was as shown in Fig. 3. As can be seen, the steady state error was approximately 0.2 ∿ 0.3V (20 ∿ 30%). The variation of the error was due to non-uniform frictions.

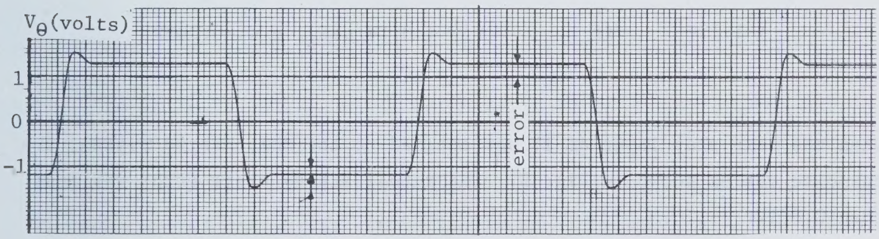


Fig. 3 STEP INPUT RESPONSE (25 mm/sec.)

To reduce the error, a microprocessor based compensation scheme was implemented as shown in Fig. 4.

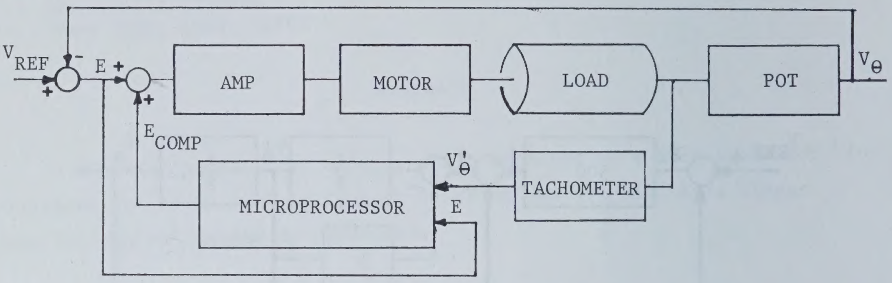


Fig. 4 SYSTEM WITH FRICTION COMPENSATION

The error and the velocity of the motor were sent to the microprocessor through A/D converters, and the compensation command E_{comp} was sent to the system input through a D/A converter. Basically, the compensation signal was determined by the following equation.

$$E_{comp} = \begin{cases} 0 & \text{if} \quad |v'_\theta| \geq 0.1\text{V/sec.} \\ 0 & \text{if} \quad |\, e \,| \leq 0.008\text{V} \\ 1.25\text{V} & \text{if} \quad |v'_\theta| \leq 0.1\text{V/sec.}, \; e > 0.008\text{V} \\ -1.25\text{V} & \text{if} \quad |v'_\theta| \leq 0.1\text{V/sec.}, \; e < -0.008\text{V} \end{cases}$$

When the same step input of 1V was applied to the compensated system, the response was as shown in Fig. 5. The actual steady state error was less than 0.001V.

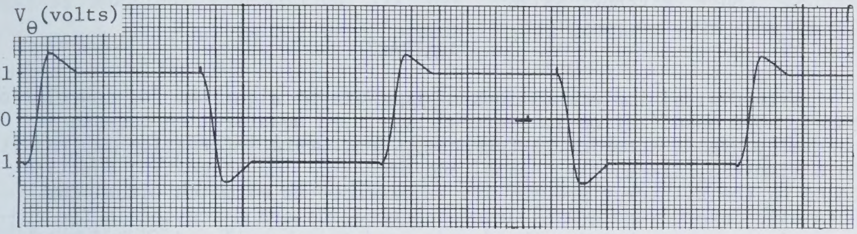


Fig. 5 STEP RESPONSE OF THE COMPENSATED SYSTEM (25 mm/Sec.)

PERFORMANCE BOUNDS FOR CONVOLUTIONAL CODES WITH MISMATCHED DIGITAL VITERBI DECODERS*

JAMES G. DUNHAM
KOU-HU TZOU
Washington University
St. Louis, Missouri 63130

ABSTRACT

With digital implementations of the Viterbi algorithm for convolutional codes, soft quantization is prefered over hard quantization because it generally yields superior performance. Inaccurate information may result in a mismatch between the channel and decoder. The generating function bound on the bit error probability is extended to mismatch situations. Automatic Gain Control (AGC) level inaccuracies are discussed in the context of a mismatch between the decoder and the channel and the mismatch bound is compared with results obtained by simulations.

I. INTRODUCTION

Short constraint length convolutional codes coupled with maximum likelihood or Viterbi decoding provide an efficient system for reliable communication on power limited satellite and space channels [1-6]. In many situations the Viterbi algorithm is implemented by sampling and quantizing the received data and performing computations with digital circuitry. With digital implementations of the Viterbi algorithm, multilevel or soft quantization is prefered over two-level or hard quantization because it generally yields superior performance [1,6,7]. However, a potential problem with soft quantization is that knowledge of the signal energy and the channel noise variance is required. If this information is inaccurate, then the Viterbi algorithm is mismatched to the channel, resulting in a potential loss in performance.

A theory which helps predict the effect of a mismatch between the channel and decoder on the bit error probability is developed. In section II the generating function approach for bounding the bit error probability is extended to mismatch situations. It is shown that a mismatch potentially increases the generating function bound and reduces to the usual bound when the channel and decoder are matched. Finally, by considering a family of related decoders, a relatively tight upper bound on the bit error probability is obtained.

Coded systems which use soft quantization require an analog to digital converter (A/D) at the modem matched filter output whose

---

*This work was supported by the National Science Foundation under grant ENG-7910001.

thresholds depend upon the correct measurement of the signal energy and channel noise variance. Since the level settings are effectively controlled by the Automatic Gain Control (AGC) circuitry in the modem, the sensitivity of the system performance to an inaccurate or drifting AGC signal is important. In section III AGC level sensitivities are studied. The mismatch bound on the bit error probability is compared with results obtained by simulation.

II. MISMATCH BOUND

A convolutional encoder is a linear finite-state machine consisting of a K-stage binary shift register and n linear algebraic functions. The binary input data is shifted along the register b bits at a time. An example with K=3, n=2 and b=1 is shown in Fig. 1. This example will be used throughout the paper.

The channel is an additive white Gaussian noise channel with binary inputs $a_0 = -a_1 = \sqrt{E_s}$, signal energy E_s and noise variance $N_o/2$. The output of the channel is symmetrically quantized to one of J+1 levels, yielding a discrete memoryless channel described by the conditional probabilities

$$p(\underset{\sim}{y}|\underset{\sim}{x}) = \prod_{n=1}^{N} p(y_n|x_n) \tag{1}$$

where for $k = 0,1$ and $j = -J/2, -J/2+1, \ldots, J/2$

$$\frac{1}{\sqrt{\pi N_0}} \int_{B_j} \exp(-(z-a_k)^2/N_0)\, dz = p(y=b_j|x=a_k) \overset{\Delta}{=} p_k(b_j) \tag{2}$$

with B_j denoting the $j^{\underline{th}}$ quantization interval and b_j being the channel output symbol corresponding to it. In particular, we note that the resultant channel is symmetric in the sense that $p_0(y) = p_1(-y)$.

The decoder is a maximum likelihood decoder. It is based upon the assumed signal energy $\hat{E}_s$ and channel noise variance $\hat{N}_0$ which yields the symmetric channel probabilities $q_0(\cdot)$ and $q_1(\cdot)$. If $p_0(\cdot) = q_0(\cdot)$, then the channel and decoder are matched. Otherwise we say that the channel and decoder are mismatched. As is well known, the decoder can be implemented efficiently by a Viterbi algorithm based upon the path metric

$$\prod_{n=1}^{N} q(\underset{\sim}{y}_n|\underset{\sim}{x}_{mn}) \tag{3a}$$

or equivalently its logarithm

$$\ln \prod_{n=1}^{N} q(\underset{\sim}{y}_n|\underset{\sim}{x}_{mn}) = \sum_{n=1}^{N} \ln q(\underset{\sim}{y}_n|\underset{\sim}{x}_{mn}) \tag{3b}$$

where $\underset{\sim}{x}_{mn}$ is the code-subvector of the $m^{\underline{th}}$ message sequences for the $n^{\underline{th}}$ branching level.

We now derive an upper bound on the bit error probability P_b for a mismatched channel and digital Viterbi decoder. The error probability at the $j^{\underline{th}}$ node, $P_e(j)$, can be upper bounded by the probability that any path diverging from the correct path at node j accumulates higher total metric over the unmerged span of the path,

$$P_e(j) \le \Pr\left[\bigcup_{\underset{\sim}{x}_k \in X(j)} \{\Delta M(\underset{\sim}{x}_k,\underset{\sim}{x}_j) \ge 0\} \right] \tag{4}$$

where $\underset{\sim}{x}_k$ is an incorrect path diverging from the correct path at node j, $\Delta M(\underset{\sim}{x}_k|\underset{\sim}{x}_j)$ is the difference between the metric increment of the path $\underset{\sim}{x}_k$ and the correct path $\underset{\sim}{x}_j$ over the unmerged segment, and $X(j)$ is the set of all such paths. Using the union bound gives

$$P_e(j) \le \sum_{\underset{\sim}{x}_k \in X(j)} \Pr[\Delta M(\underset{\sim}{x}_k|\underset{\sim}{x}_j) \ge 0] \tag{5}$$

where each term of the summation is the pairwise error probability for two code vectors over the unmerged segment.

Using the memoryless property of the channel,

$$\begin{aligned}\Pr[\Delta M(\underset{\sim}{x}_k,\underset{\sim}{x}_j) \ge 0] &= \sum_{\{\underset{\sim}{y}|q(\underset{\sim}{y}|\underset{\sim}{x}_k) \ge q(\underset{\sim}{y}|\underset{\sim}{x}_j)\}} p(\underset{\sim}{y}|\underset{\sim}{x}_j) \\ &\le \sum_{\underset{\sim}{y}} p(\underset{\sim}{y}|\underset{\sim}{x}_j) \left[q(\underset{\sim}{y}|\underset{\sim}{x}_k)/q(\underset{\sim}{y}|\underset{\sim}{x}_j)\right]^{1/2} \\ &= \left[\sum_{y} p_0(y)[q_1(y)/q_0(y)]^{1/2}\right]^{w_k} \end{aligned} \tag{6}$$

where w_k denotes the number of positions in which $\underset{\sim}{x}_j$ and $\underset{\sim}{x}_j$ differ and we assume that the all-zeros path is the correct path. Letting

$$c(y) = \left[\frac{q_1(y)/p_1(y)}{q_0(y)/p_0(y)}\right]^{1/2} , \tag{7}$$

we obtain using the symmetry of the channel

$$\sum_y p_0(y)[q_1(y)/q_0(y)]^{1/2} = \sum_y c(y)\sqrt{p_0(y)p_1(y)}$$

$$= \sum_y (c(y) + c(y)^{-1})\sqrt{p_0(y)p_0(-y)} + p_0(0).$$

$$\triangleq Z\ (q||p)\ . \tag{8}$$

When the channel and decoder are matched, $q_0(y) = p_0(y)$ and

$$Z(p||p) = \sum_y \sqrt{p_0(y)p_1(y)}\ , \tag{9}$$

the usual result [7,Eq.(2.9.19)]. Since $x + x^{-1} \geq 2$ for all $x > 0$, it follows from (8) that

$$Z(q||p) \geq Z(p||p)\ . \tag{10}$$

Thus the effect of a mismatched channel and decoder is to increase the pairwise error probability bound (4). We point out that $Z(p||p)$ is known as the Bhattacharyya bound, and so $Z(q||p)$ can be viewed as a generalization of it to mismatch situations.

Letting $T(D)$ and $T(D,I)$ denote the generating function and augmented generating function, respectively, we obtain as in [7,Eq.(4.4.1)-(4.4.13)]

$$P_e(j) \leq T(D)\Big|_{D=Z(q||p)} \tag{11}$$

and

$$P_b(j) \leq b^{-1}\frac{\partial T(D,I)}{\partial I}\Bigg|_{I=1\ ,\ \ D=Z(q||p)} \triangleq P_b(q||p)\ . \tag{12}$$

Since $T(D)$ and $\partial T(D,I)/\partial I$ are both nondecreasing functions in D, we see from (10) that the overall result of a mismatched channel and decoder is to potentially increase the upper bound on P_b from $P_b(p||p)$ to $P_b(q||p)$. If the mismatch is small, then both bounds will be about the same. This indicates that the bit error probability for the mismatched situation will be with bounds of the match situations.

We observe that the Viterbi algorithm uses the path metrics $d(\underset{\sim}{y}|\underset{\sim}{x})$ where for some $\alpha > 0$ and β

$$\alpha d(\underset{\sim}{y}|\underset{\sim}{x}) + \beta = -\ln q(\underset{\sim}{y}|\underset{\sim}{x}) \tag{13}$$

and seeks the path through the trellis which minimizes

$$d(\underset{\sim}{y}|\underset{\sim}{x}) = \sum_{n=1}^{N} d(\underset{\sim}{y}_n|\underset{\sim}{x}_{mn}) \tag{14}$$

or, equivalently, the path which maximizes (3b) for the received sequence y.

For $\alpha > 0$

$$q_\alpha(y|x) = \frac{\exp\left(-\alpha d(y|x)\right)}{\sum_z \exp\left(-\alpha d(z|x)\right)} \tag{15}$$

is a probability measure where

$$\begin{aligned} \ln q_\alpha(y|x) &= -\alpha d(y|x) - \ln \sum_z \exp\left(\alpha d(z|x)\right) \\ &= -\alpha d(y|x) + \beta. \end{aligned} \tag{16}$$

Since the constants α and β are unimportant in finding the path through the trellis of minimal total metric, we see that the decoder is matched to a channel of the form $q_\alpha(y|x)$ for arbitrary $\alpha > 0$ while the real channel is $p(y|x)$. Thus

$$\overline{P}_b(q||p) \triangleq \inf_{\alpha>0} P_b(q_\alpha||p) \tag{17}$$

is a relatively tight bound on the bit error probability.

III. AGC MISMATCH

We model an inaccurate AGC level as shown in Fig. 2. The modem matched filter output is a Gaussian random variable X with mean $\sqrt{2S}$ and variance 1 where S denotes the signal-to noise ratio. The inaccuracies in the AGC signal level are modeled as an amplifier with gain A that produces the signal $\hat{X} = AX$ which is entered into the A/D converter. Thus the A/D actually sees a Gaussian random variable with mean $A\sqrt{2S}$ and variance A^2. The Viterbi decoders assumes that it is viewing the signal X when in fact it is viewing the signal $\hat{X}$.

As is well known, the Viterbi decoder does not depend upon the actual signal-to noise ratio S for both hard and infinitely fine quantization level cases. Thus AGC level inaccuracies are relatively unimportant in these two cases. However, for soft quantization, the accuracy of the AGC level is important and can cause degradation in system performance [1]. Thus we are interested in obtaining bounds on how much A can change before it begins to significantly affect system performance.

The convolutional code of Fig. 1 was used with the 8-level uniform quantizer shown in Fig. 3. The results of a simulation of P_b using 10^6 samples is shown in Figs. 4 and 5 along with the mismatch bound $\overline{P}_b(q||p)$ for signal-to-noise ratios of 4 and 6 db, respectively. From both figures we see that the mismatch bound gives a good indication of the sensitivity of P_b to AGC level inaccuracies.

IV. CONCLUSIONS

The transfer function bound on the bit error probability for convolutional codes has been extended to situations in which the Viterbi decoder is mismatched to the channel. The mismatch bound can be used to help design digital implementations of the Viterbi algorithm since it provides easily computed analytic bounds on their performance. In particular, the mismatch bound has been applied to the problem of AGC level sensitivities.

REFERENCES

[1] J.A. Heller and I. M. Jacobs, "Viterbi Decoding for Satellite and Space Communication," IEEE Trans. Commun. Technol., vol. COM-19, pp. 835-848, October 1971.

[2] A. J. Viterbi, "Convolutional Codes and Their Performance in Communication Systems," IEEE Trans. Commun. Tech., vol. COM-19, pp. 751-772, October 1971.

[3] A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymplatic Optimum Decoding Algorithm," IEEE Trans. Inform. Theory, vol. IT-13, pp. 260-269, April 1967.

[4] I. M. Jacobs, "Sequential Decoding for Efficient Communication from Deep Space," IEEE Trans. Commun. Tech., vol. COM-15, pp. 492-501, August 1967.

[5] J. A. Heller, "Short Constraint Length Convolutional Codes," Jet Propulsion Lab., California Inst. Technol., Space Program Summary 37-54, vol. III, pp. 171-177, Oct./Nov. 1968.

[6] Linkabit Corp., Final Rep., "Coding Systems Study for High Data Rate Telemetry Links," Contract NAS2-6024, NASA Ames. Res. Ctr. Rep. CR-11427B. Moffett Field, California.

[7] A. J. Viterbi and J. K. Omura, Principles of Digital Communication and Coding. New York: McGraw-Hill, 1979.

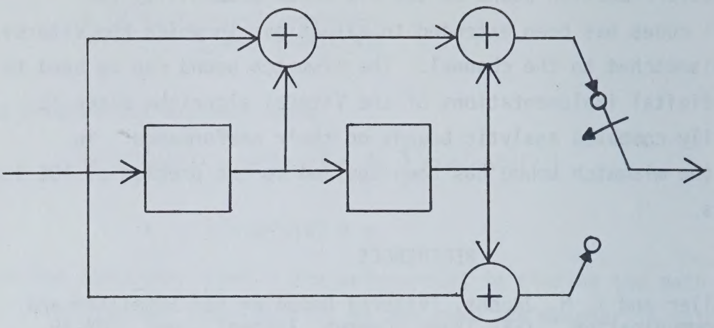


Fig. 1. Convolutional Coder for K = 3, n = 2 and b = 1.

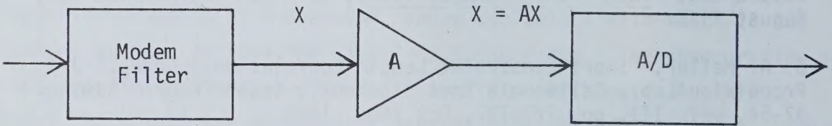

X
$\hat{X} = AX$
Modem Filter
A
A/D

Fig. 2. Model for AGC Mismatch.

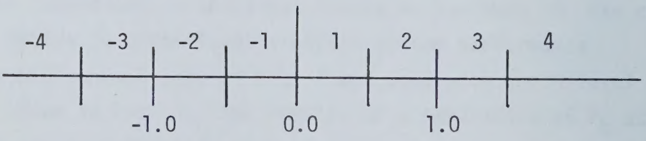

-4
-3
-2
-1
1
2
3
4
-1.0
0.0
1.0

Fig. 3. Receiver Quantization Thresholds and Intervals for 8-Level Quantization.

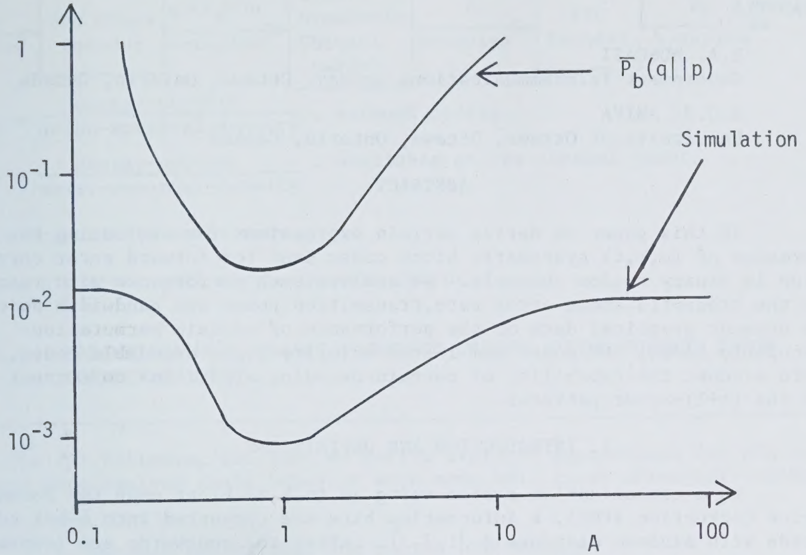


Fig. 4. Bit Error Probability for AGC Mismatch: S/N = 4 dB.

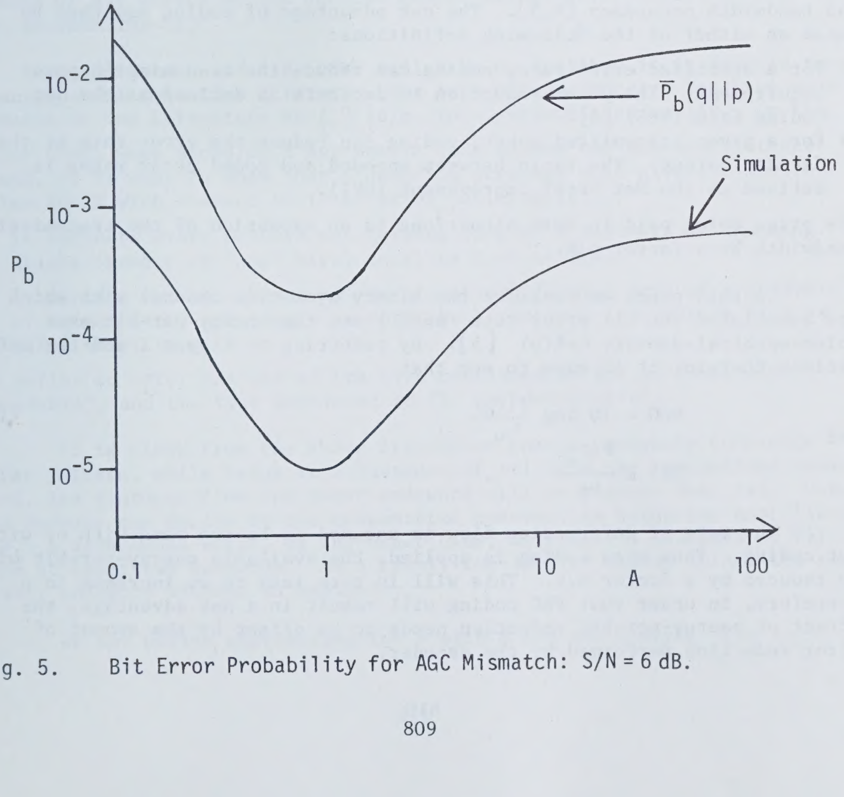


Fig. 5. Bit Error Probability for AGC Mismatch: S/N = 6 dB.

ON THE PERFORMANCE OF CERTAIN BLOCK ERROR-CORRECTING CODES IN BINARY CHANNELS

Z.A. MUSCATI
Government Telecommunications Agency, Ottawa, Ontario, Canada

S.G.S. SHIVA
University of Ottawa, Ottawa, Ontario, Canada

ABSTRACT

In this paper we derive certain expressions for evaluating the performance of (n,k,t) systematic block codes used for forward error correction in binary random channels. We analyze such performance with respect to the tradeoffs among error rate,transmitted power and bandwidth occupancy. We present graphical data on the performance of certain permutation-decodable codes, BCH codes and 1-step majority-logic decodable codes, taking into account the capability of certain decoding algorithms to correct some of the (t+1)-error patterns.

1. INTRODUCTION AND DEFINITIONS

In a communication system using an (n,k,t) block code for Forward Error Correction (FEC), k information bits are converted into n-bit codewords with minimum distance d [1,2,3]. After the codewords are transmitted through a noisy channel, a decoder can usually correct t errors or fewer in every received block of n bits, where t is the integer part of (d-1)/2.

In many practical applications of FEC coding, it is required to consider its effects on the tradeoffs among the error rate, transmitted power and bandwidth occupancy [4,5]. The net advantage of coding may then be based on either of the following definitions:

a) For a specified error rate, coding can reduce the transmission power requirement. The power reduction in decibels is defined as the Net Coding Gain (NCG) [4].
b) For a given transmitted power, coding can reduce the error rate at the decoder output. The ratio between uncoded and coded error rates is defined as the Net Error Improvement (NEI).

The price being paid in both situations is an expansion of the transmission bandwidth by a factor n/k.

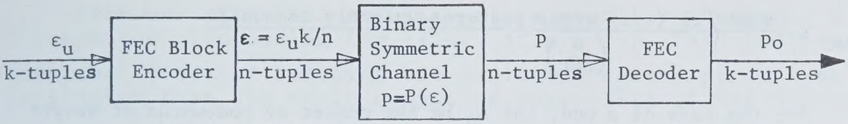
In this paper we consider the binary symmetric channel with which are associated the bit error rate $p=P(\varepsilon)$ and the energy-per-bit over noise-spectral-density $\varepsilon=\xi(p)$ [5]. By referring to Figure 1 and the definitions therein, it is easy to see that

$$NCG = 10 \log \frac{\xi(p_o)}{\varepsilon_u} \tag{1a}$$

and

$$NEI = \frac{P(\varepsilon_u)}{p_o} \quad . \tag{1b}$$

The rate of information flow is assumed to be the same with or without coding. Thus when coding is applied, the available energy-per-bit will be reduced by a factor n/k. This will in turn lead to an increase in p. Therefore, in order that FEC coding will result in a net advantage, the effect of energy-per-bit reduction needs to be offset by the amount of error reduction performed by the decoder.



$\varepsilon_u = \dfrac{\text{energy-per-bit}}{\text{noise-spectral-density}}$, without coding.

$\varepsilon = \dfrac{\text{energy-per-bit}}{\text{noise-spectral-density}}$, available at the channel input.

p = channel bit error rate.

p_o = bit error rate at the decoder output.

FIGURE 1: PARAMETERS OF A BINARY SYMMETRIC CHANNEL USING FORWARD ERROR CORRECTION (FEC).

In the following sections we derive explicit expressions for NCG and NEI, and then analyze their behavior with some well known error-correcting codes.

2. GENERAL DERIVATIONS FOR CODING PERFORMANCE

For the case of binary symmetric channel with bit error rate $p \ll 0.1$ the decoder performance is dominated by the occurance of (t+1)-error patterns and will depend on the value of N. Here N is defined as the mean number of bit errors at the decoder output due to (t+1)-error patterns at the decoder input.

A distance-bounded decoder will take no decision if it detects the occurrence of more than t errors. For such circumstances N has been upper-bounded in the literature by $\binom{n}{t+1}d/n$, for systematic codes [4]. On the other hand, when decoding a systematic code beyond the minimum distance bound, it is easy to show that a decoding strategy will give the smallest value for N with respect to (t+1)-error patterns if:

a) It corrects every pattern not forming part of a codeword of Hamming weight (number of "one" bits) equal to 2t+1 or 2t+2.

b) It decides not to correct any error pattern forming part of a codeword of weight 2t+2, and simply accepts the k received information bits as being error free.

We define an error pattern of the type mentioned in (a) as "uniquely decodable", and the type mentioned in (b) as "detectable".

It is clear from the above discussion that a uniquely decodable error pattern, while being at a distance of t+1 from the transmitted codeword, its distance from any other codeword will be greater than t+1. Hence the decoder can decide on the transmitted codeword as being the most likely one. A detectable error, on the other hand, will have a distance of t+1 from at least two codewrds. As a result the decoder cannot select the most likely codeword without an ambiguity.

We now derive expressions for N with d odd and d even. Let

$$Q_{t+1} = \frac{\text{Number of (t+1)-error patterns uniquely decodable}}{\binom{n}{t+1}} \qquad (2)$$

For the case of d odd, let A_d be the number of codewords of weight $d = 2t+1$. When a (t+1)-error pattern forming part of a codeword of weight d occurs, the decoder will erroneously indicate an error pattern of weight t causing 2t+1 errors at the output. When other (t+1)-error patterns occur that are detectable but not uniquely decodable, the decoder will take no decision with the result of t+1 errors remaining at its output. On the basis of this we obtain,

RESULT 1a: For d odd, $N = N_1 = \frac{t\,A_d}{n}\binom{2t+1}{t} - \binom{n-1}{t} - \frac{(t+1)\,Q_{t+1}}{n}$.

For the case of d even, on the other hand, all (t+1)-error patterns are either uniquely decodable giving zero errors, or detectable giving t+1 errors in the n-tuple at the decoder output. On this basis we obtain,

RESULT 1b: For d even, $N = N_2 = \binom{n-1}{t} - \frac{t+1}{n}\,Q_{t+1}$.

Expressions for N with distance-bounded decoding are directly attainable from Results 1a and 1b by setting $Q_{t+1} = 0$.

Once the value of N is determined for a given (n,k,t) code, the bit error rate p_o at the decoder output due to an error rate p at the input is given by

$$p_o = p^{t+1}\,(1-p)^{n-t-1}\,N \quad , \text{ for } p \ll 0.1 \quad . \qquad (3)$$

By referring to Figure 1 and combining (3) with (1a) and (1b) respectively, we obtain explicitly the following expressions

$$\mathrm{NCG} = 10\,\log\,\frac{\xi\{(P(\frac{k}{n}\varepsilon))^{t+1}\,(1-P(\frac{k}{n}\varepsilon))^{n-t-1}\,N\}}{\varepsilon} \quad , \qquad (4)$$

and

$$\mathrm{NEI} = \frac{P(\varepsilon)}{\{P(\frac{k}{n}\,\varepsilon)\}^{t+1}\,\{1-P(\frac{k}{n}\,\varepsilon)\}^{n-t-1}\,N} \quad . \qquad (5)$$

On the basis of these expressions we now define the cut-off error rate p_{co} to be such that, for a given (n,k,t) code, there is a coding advantage if $p < p_{co}$. Setting NEI = 1 in (5), the value of p_{co} can then be obtained by solving

$$p_{co} = \{P(\frac{k}{n}\,\xi(p_{co}))\}^{t+1}\,\{1-P(\frac{k}{n}\,\xi(p_{co}))\}^{n-t-1}\,N \quad . \qquad (6)$$

From our numerical results, it is seen that there is a net degradation with coding for values of $p_{co} > p$. This indicates that the effect of reducing the available energy-per-bit by n/k is greater than the error-reduction capability of the decoder.

We now examine the performance of coding with the binary white Gaussian channel with or without PSK, FSK or ASK modulation and using either coherent or non-coherent detection [6]. The relationship between p and ε for each of these cases is given in the Appendix. In all these cases the coding advantage is seen to be greater for smaller error rates. By using the approximation

$$\operatorname{erfc}(x) \simeq \frac{\exp(-x^2)}{x\sqrt{\pi}}, \text{ for large } x, \tag{7}$$

and combining (A1) to (A4) with (4) respectively, we can take the limit of NCG with increasing ε. We then find that as p approaches zero, NCG approaches an upper bound given by

$$Bg = 10 \log \frac{(t+1)k}{n}. \tag{8}$$

3. GRAPHICAL RESULTS ON CODING PERFORMANCE

The upper bound Bg on the coding gain serves as a simple and channel-independant criterion in comparing the potential coding advantage of different (n,k,t) codes without the knowledge of their corresponding values of A_d or Q_{t+1}. Bg is especially useful in discarding codes of relatively little coding advantage in the process of selecting codes for improving the channel performance. For example, Bg is found to be negligibly small or even to have negative values for several short codes of low coding ratio k/n which are known to be "optimum" [1,7] in terms of their (n,k,t) parameters, e.g. the m-sequence codes ($n=2^m-1$, $k=m$, $t=2^{m-2}-1$), the (15,7,2) BCH code and the (7,4,1)Hamming code, to cite only a few cases.

Figure 2 plots Bg versus bandwidth expansion factor n/k for some easily decodable codes which yield positive values for Bg. For the case of"permutation-decodable" codes [2,3] we have $t \le \frac{n}{k}$ [2] giving $Bg \le 10 \log(1+\frac{k}{n})$ which provides an upper limit on Bg independant of the value of t.

In order to present some illustrative graphical results on NCG, NEI and p_{co} we choose the case of non-coherent FSK because of its analytical simplicity and practical significance. Combining (A3) with (4) and (5) respectively gives

$$NCG = 10 \log \left(\frac{k(t+1)}{n} - \frac{(t)\log 2 - \log (N) - (n-t-1)\log(1-0.5(2p)^{k/n})}{\log(2p)} \right) \tag{9}$$

and

$$NEI = \frac{2^t (2p)^{1-k(t+1)/n}}{[1 - 0.5(2p)^{k/n}]^{n-t-1} \, N}. \tag{10}$$

Assuming that $np \ll 1$, (9) and (10) can be respectively approximated to

$$NCG \simeq 10 \log \left[\frac{k(t+1)}{n} - \frac{(t)\log 2 - \log N}{\log(2p)}\right], \tag{11}$$

and $$NEI \simeq \frac{2^t}{N}(2p)^{1-(t+1)k/n}. \tag{12}$$

For the case of BCH codes we assume a distance-bounded decoder [8], hence setting $Q_{t+1}=0$. For d even, therefore, N_2 can be readily estimated from Result 1b. For d odd, the value of A_d is known for d=3,5,7 [8] which can be applied in Result 1a to obtain N_1.

In investigating cyclic codes which are majority-logic-decodable in one step [9,10], we found that going through two rounds of error estimation would significantly increase the value of Q_{t+1}. This fact has been taken into account when estimating the results given in this section.

Figure 3 plots the cut-off error rate (p_{co}) versus n/k. This value roughly indicates the range of channel error rates p at which NCG will be close to its upper bound Bg, i.e. when $p \ll p_{co}$. It is noted that none of the codes plotted in Figure 3 offers an advantage at $p \geq 5 \times 10^{-3}$.

Figure 4 plots NCG and NEI versus n/k at $p = 10^{-5}$. In this figure the same plots serve to represent NCG and NEI respectively by using two different vertical scales. This is because there is a direct correspondance between these two parameters for a given p as can be seen by combining (11) with (12) to get

$$NCG = 10 \log \left(1 - \frac{\log(NEI)}{\log(2p)}\right). \tag{13}$$

It is worth noting from Figures 2 and 4 that the coding advantage in BCH codes will reach an optimum point when increasing t for a given n or decreasing n for a given t. Beyond such an optimum point, there is no performance improvement in further expanding the bandwidth by n/k. This behavior clearly illustrates the two counteracting effects of coding mentioned in Section 1.

APPENDIX

ERROR RATES IN BINARY WHITE GAUSSIAN CHANNELS

The following is a list of relationships between the channel error rate p and the energy-per-bit over noise-spectral-density ε for various binary white Gaussian channels with and without different carrier modulation schemes [6].

I. $$p = \frac{1}{2}\,\mathrm{erfc}\sqrt{\varepsilon}, \tag{A1}$$

for polar baseband (non-return-to-zero) and for phase-shift keying (PSK) with coherent detection.

II. $$p = \frac{1}{2}\,\mathrm{erfc}\sqrt{\frac{\varepsilon}{2}}\,, \qquad \text{(A2)}$$

for amplitude-shift keying (ASK) with coherent detection and for frequency-shift keying (FSK) with coherent detection.

III. $$p = \frac{1}{2}\,\exp(-\frac{\varepsilon}{2}), \qquad \text{(A3)}$$

for FSK with noncoherent detection.

IV. $$p \simeq \frac{1}{2}\;1+(2\pi\varepsilon)^{-1}\;\exp(-\frac{\varepsilon}{2})\,, \qquad \text{(A4)}$$

for ASK with noncoherent detection.

ACKNOWLEDGEMENT

The work reported in this paper was supported in part by the National Research Council of Canada under Grant Number A-3371.

REFERENCES

[1] F.J. MacWilliams and N.J.A. Sloane, The Theory of Error-Correcting Codes, Amsterdam, Netherlands: North-Holland, 1977.

[2] W.W. Peterson and E.J. Weldon, Error-Correcting Codes, 2nd ed. Cambridge, MA: MIT Press, 1972.

[3] S. Lin, An Introduction to Error-Correcting Codes, Englewood Cliffs, N.J.: Prentice-Hall, 1970.

[4] J.H. Blythe and K. Edgecombe, Net coding gain of error-correcting codes, Proc. IEE, Vol. 122, No. 6, pp. 609-614, June 1975.

[5] I.M. Jacobs, Practical applications of coding, IEEE Trans. Inform. Theory, Vol. IT-20, No. 3, pp. 305-310, May 1974.

[6] Z.P. Peyton, Jr., Communication System Principles, Reading MA: Addison-Wesley, 1976.

[7] H.J. Helgert and R.D. Stinaff, Minimum-distance bounds for binary linear codes, IEEE Trans. Inform. Theory, Vol. IT-19, pp. 344-356, May 1973.

[8] E.R. Berlekamp, Algebraic Coding Theory, New York: McGraw-Hill, 1968.

[9] E.J. Weldon, Difference-set cyclic codes, B.S.T.J., pp. 1045-1055, September 1966.

[10] C.L. Chen and W.T. Warren, A note on one-step majority-logic decodable codes, IEEE Trans. Inform. Theory, Vol. IT-19, pp. 135-137, January 1973.

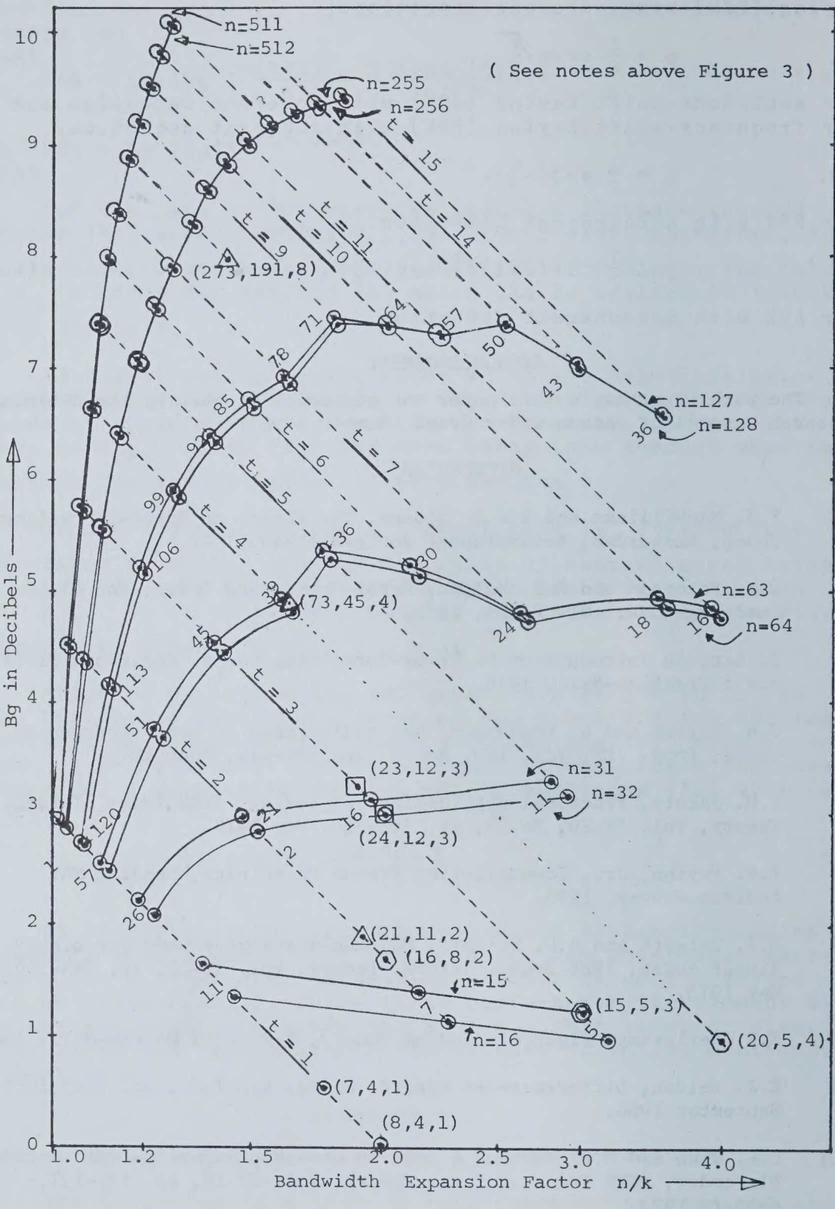


FIGURE 2: UPPER BOUND (Bg) ON CODING GAIN VERSUS BANDWIDTH EXPANSION FACTOR n/k

Notes on Figures 2, 3 and 4:-

Single numbers indicate values of k.

- ⊙ BCH Codes.
- ◎ Permutation-Decodable Codes.
- ◬ Difference-Set Codes.
- ⊡ Golay Code.

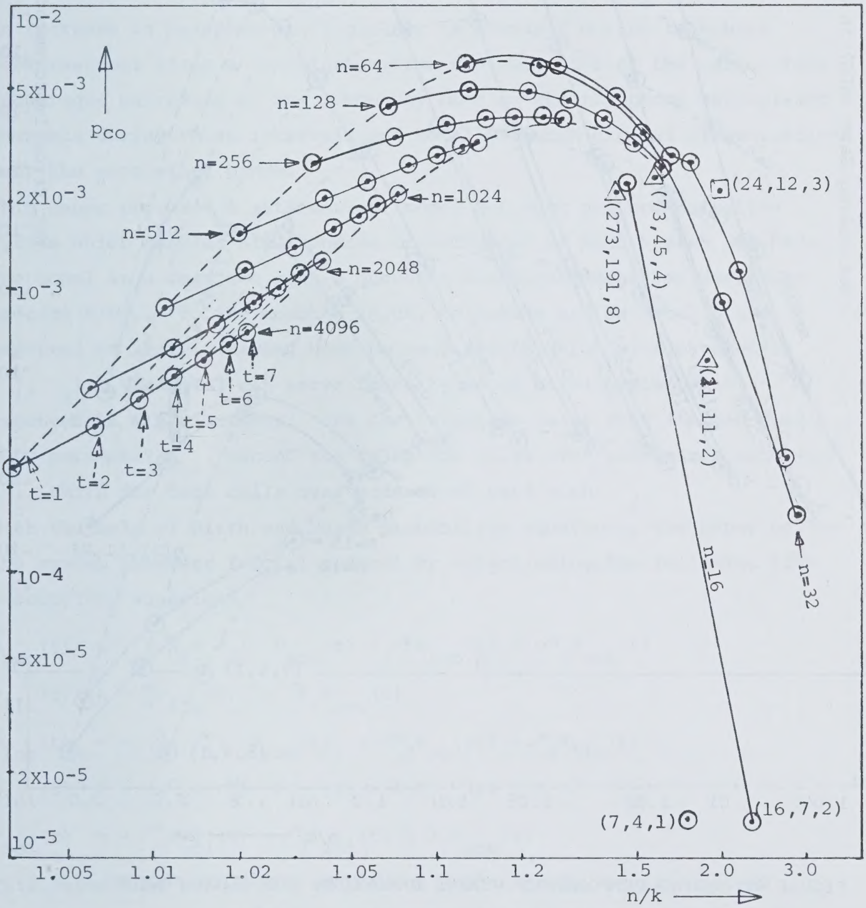


FIGURE 3: CUT-OFF ERROR RATE VERSUS BANDWIDTH EXPANSION FACTOR FOR BINARY NON-COHERENT FSK

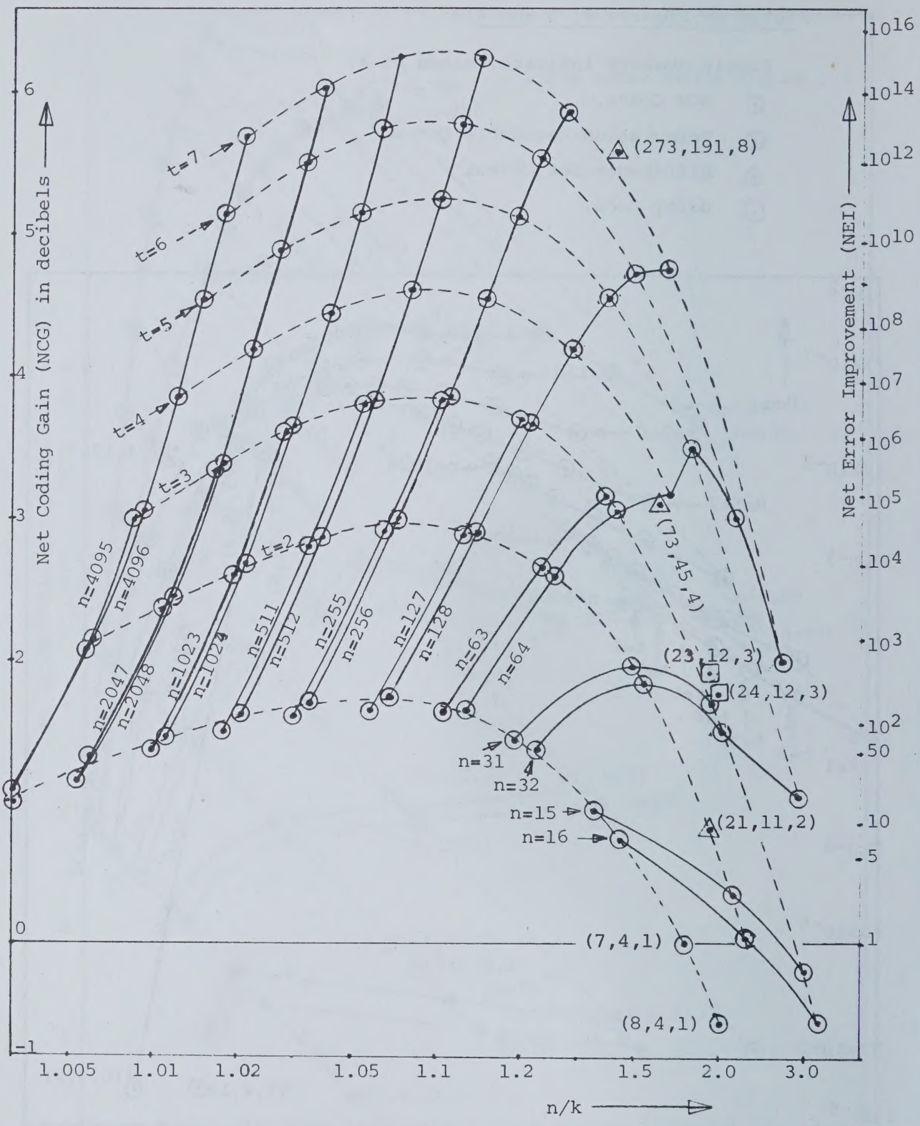


FIGURE 4: CODING PERFORMANCE VERSUS BANDWIDTH EXPANSION FACTOR FOR BINARY NON-COHERENT FSK AT $p = 10^{-5}$

MATHEMATICAL MODEL FOR A NEW COMMUNICATION SYSTEM ENABLING SIMULTANEOUS TRANSMISSION OF SPEECH AND DATA

DR. ENG. M. NAGI ELYOUSFI AND DR. ENG. M. MUSTAFA RATEB

Chief of Communication Department, Military Technical College,
Cairo, Egypt.

Summary

An increase in telephone call quality is achieved during telephone conversations after a few minutes from the beginning of the call. This encourages narrowing of the frequency band of the telephone multiplexed channels during these intervals and to allot extra digital communications over the economised bands.

This paper proposes a mathematical model for such new communication system which enables simultaneous transmission of both speech and data. The model is a loss one with a queueing stochastic behavior marked by special M/M/C. To the model's input, telephone and data calls are expected to arrive (having both Poisson distribution with parameters $_1$, $_2$). The model can serve three types of service-time with exponential distributions: one for telephone calls over limited band with parameter $_1$, second for telephone calls over economised band with $_2$, third for data calls over economised band with $_3$.

With the help of Birth and Death probability equations, the paper solves the system behavior for C=1 channel by establishing the following five probability equations:

$$\dot{P}_{ooo}(t) = -(\lambda_1+\lambda_2)\,P_{ooo}(t) + \mu_1 P_{1oo}(t) + \mu_3 P_{oo1}(t)$$

$$\dot{P}_{11o}(t) = -\mu_2 P_{11o}(t) + \lambda_1 P_{ooo}(t)$$

$$\dot{P}_{1oo}(t) = -(\lambda_2+\mu_1)\,P_{1oo}(t) + \mu_3 P_{1o1}(t) + \mu_2 P_{11o}(t)$$

$$\dot{P}_{1o1}(t) = -(\mu_1+\mu_3)\,P_{1o1}(t) + \lambda_2 P_{1oo}(t)$$

$$\dot{P}_{oo1}(t) = -\mu_3 P_{oo1}(t) + \mu_1 P_{1o1}(t) + \lambda_2 P_{ooo}(t)$$

This paper then solves the system for C=2 channels, where fourteen equations are in question and their solutions are laborious. But for C=24 channels the system solution is made by computer-simulation where the significant results obtained show the capability of the proposed system to serve both speech and data with negligible probability of loss. This proves the advantage of the system's introduction.

BOUNDS ON MESSAGE START INFORMATION FOR MESSAGES SUBJECT TO QUEUEING DELAY

K.M. MACKENTHUN JR.
University of South Carolina
Columbia, SC 29208

The design of efficient low overhead protocols is an important consideration in the optimization of network performance. Gallager showed that this problem could be approached from an information theory setting and gave a lower bound on the amount of message start information in a network with average message delay d [1]. Gallager studied networks in which queueing delays could be neglected; in this paper, we extend Gallager's discussion to networks with queueing delays. Specifically, we show that Gallager's lower bound is tight for networks with queueing delay, and we construct protocols that use close to the optimum amount of protocol start information per message.

I. INTRODUCTION

Traffic in a data network appears in two distinct forms: message traffic, and an overhead traffic, often called protocol information, which controls the flow of messages within the network and ensures that they are received correctly at the proper destination. Many networks use more protocol information than message information; in these networks especially it is important to use low overhead protocols to allow as much of the network capacity as possible to be used for message information. The problem of finding the least amount of protocol start/stop information for a message in a network was first discussed in a nice article by Gallager [1]. The basic question that Gallager addresses is how many bits/message are required to transmit protocol start/stop information from source to receiver subject to an average message delay d; in [1], Gallager gives a lower bound for this protocol information.

Consider a source in the network that is sending a sequence of messages to a receiver at another node. In each time unit, the receiver makes a binary decision on the received signal. In order for the receiver to abstract messages from the binary stream, it must know the message start time and message stop time, or equivalently, the message start time and message length. The average number of bits/message required to transmit the message length is simply the entropy of the message length.

To study the average number of start bits/message required, consider the source-receiver pair modelled in Figure 1a. Let X_i be the message arrival time of the i-th message at the source and Y_i the message arrival time at the receiver. We assume that the messages have no length and arrive instantaneously at the source. We assume that the source process is a Poisson process of rate λ. We assume that the communication link introduces no transmission delay. For any given network and network protocol, and for any given integer n, there will be a joint probability measure P_n on $X^n=(X_1,\ldots,X_n)$ and $Y^n=(Y_1,\ldots,Y_n)$. There are two obvious constraints on the probability measure P_n; first, the marginal distribution of X^n must satisfy the Poisson process assumption, and second, for each message the delay Y_i-X_i must be nonnegative with probability one. For a given P_n, there are two quantities of interest: an average mutual information between the sequence X^n and Y^n, $I_{P_n}(X^n;Y^n)$; and an expected delay per message, $E\Sigma(Y_i-X_i)/n$. Define $P_n(d)$ as the class of probability measures satisfying the above constraints and having an expected delay per message $\leq d$. The n-th order lower bound on the amount of transmitted protocol information, subject to an expected delay $\leq d$, is defined by

$$R_n(d) = \inf\left\{\frac{1}{n}I_{P_n}(X^n,Y^n) \mid P_n \varepsilon P_n(d)\right\}. \quad (1)$$

Gallager then defines

$$R(d) = \lim_{n\to\infty} \inf R_n(d). \quad (2)$$

and proves the following theorem.

Theorem (Gallager). $R_1(d)$, as given by (1) with n=1, is a lower bound to $R_n(d)$ for all $n > 1$, to $R(d)$, and to the average protocol information per message about message arrival times between a source-receiver pair for Poisson message arrivals of rate λ and expected delay d. Furthermore, $R_1(d)$ is given by

$$R_1(d) = -\log_2(1-e^{-\lambda d}) \text{ bits/message.} \quad (3)$$

The probability measure P_1 that achieves $R_1(d)$ is defined implicitly by

$$Y_1=\max(X_1,d) + Z \quad (4)$$

where Z is a random variable independent of X_1 with density

$$P_2(z) = \begin{cases} ce^{-cz} & , z \geq 0 \\ 0 & , z < 0 \end{cases}$$

where

$$c = \frac{\lambda}{1-e^{-\lambda d}}$$

As pointed out by Gallager, it is not clear whether any protocols exist which come close to the lower bound. One of the striking features of [1] is that Gallager constructs several simple protocols which do come close to the lower bound.

In a practical network, messages will be delayed as they queue for transmission over the network links. In this paper, we want to study the effect of queueing delay on the amount of protocol start information/message when the average message delay is no greater than d. A formal extension of Gallager's minimization problem can be made for situations where a queue forms between the source and receiver, as shown in Figure 1b. Because of space limitation, this discussion is not presented here. A measure can be taken from the constraint set of this minimization and a calculation of $I(X_1;Y_1)$ and average delay made. The results of this calculation are compared with $R_1(d)$ in Figure 2; Figure 2 illustrates that

$R_1(d)$ is a tight lower bound for source receiver pairs communicating through an intervening gueue. In Section II, we discuss a realistic network model in which several sources use the same channel to communicate with several receivers. For this network, we discuss several protocols which can be shown to have performance close to $R_1(d)$.

SECTION II. MESSAGE START PROTOCOLS

Consider the network shown in Figure 3 which may represent a particular network, a segment of a larger network, or a group of interactive terminals accessing a common processor. As shown in the figure all source-receiver pairs use the same channel or more generally the same service facility to communicate. At the input end, messages are multiplexed together for transmission over the channel. We assume that the line between the multiplexor and demultiplexor is synchronous; the lines feeding the multiplexor may be asynchronous. The sources are assumed to be identical independent Poisson processes of rate λ. We henceforth assume for the greatest relevance that messages in the network are of fixed length, $\bar{x}$. We define the traffic intensity $\rho=n\lambda\bar{x}$. As shown in the figure, each source S_i is assumed to be paired with a receiver R_i. As mentioned by Gallager this is no restriction since in any network, we may always visualize any source as paired with a virtual receiver or any receiver as paired with a virtual source. For example, consider the situation where several sources wish to communicate with the same receiver as in a collection of interactive terminals that access the same processor. In this case, the processor has an address location for each terminal accessing it, and we would model the processor as several virtual receivers, each paired with a terminal.

We now consider a simple protocol for the network of Figure 3 which we call the pure addressing protocol. When there is no message in the channel input queue, the input processor transmits an idle bit, for example a 0, over the channel. When the input queue is not empty, the processor takes a message from the queue, prefixes the message with the appropriate $\lceil \log_2 n \rceil$ bit source address and a start bit, for example a 1, and then inserts this bit sequence into the channel stream. For this protocol, an analytical calculation of the number of protocol bits/message and the average message delay is easy. Assume each message contains k bits of information. For the present discussion we consider the message length to include the protocol information so that $\bar{x}$ is given by

$$\bar{x} = k + \lceil \log_2 n \rceil + 1.$$

For the M/D/1 queue, the average message delay d is given by

$$d = \frac{\rho}{2(1-\rho)}\,\bar{x}$$

so that

$$\lambda d = \frac{\rho^2}{2n(1-\rho)}.$$

We now calculate the number of protocol bits/message. The average number of idle bits in 1 second is $p_o C$ where p_o is the probability that the input queue is empty and C is the channel capacity. The average number of messages in 1 second is λ. Then the number of idle bits/message is

$$\frac{p_o C}{\lambda} = \frac{p_o \bar{x}}{\left(\frac{\lambda \bar{x}}{C}\right)} = \left(\frac{1-\rho}{\rho}\right) \bar{x} \tag{5}$$

This gives the number of protocol bits/message as

$$1 + \lceil \log_2 n \rceil + \left(\frac{1-\rho}{\rho}\right) \bar{x} \tag{6}$$

It is evident from (6) that the number of protocol bits/message grows in direct proportion to $\bar{x}$, the message length, so that for reasonable message lengths, the performance curve lies well above the lower bound. The problem of course is that while the pure addressing protocol involves almost no protocol induced delay, the protocol information is very large because the input processor must transmit idle bits when there are no messages in in the input queue. To prevent this, we must queue the messages for transmission at longer time intervals. Of course such queueing involves a protocol induced delay not present in the pure addressing scheme, but as we will now see such delay can be utilized to decrease the amount of protocol information.

We now describe a protocol which approaches the lower bound on the amount of protocol information/message. In the protocol, time is divided into frames of length T, and the transmission of messages is synchronous with the start of a frame. The length of the frame is adjusted to give the required message delay; as T increases, the message delay becomes larger. The protocol requires that the channel input processor and channel output processor keep a list of messages in the queue; associated with each message in the list is the message address. At the beginning of a frame, the input processor samples each source queue, removing messages that have arrived during the previous frame and updating its list. During the first part of the succeeding frame, the input processor sends protocol information which allows the output processor to update its own list; during the second part of the frame, the input processor takes messages in order from the top of the list and places them in the frame. Upon receipt of a message, the output processor checks its own list for the message address and then routes the message to the correct destination. The order in which the input processor composes the list of messages depends on the priority assigned to each source. It is clear that as the priority structure changes, the average message delay for a particular source changes, while by the conservation principle for priority queues, the message delay averaged over all messages does not change. One simple priority would be to sample the sources in numerical order; clearly in such a scheme messages from sources lower in the list would have less average delay than messages from sources higher in the list. If it was desired to equalize the average message delay for each source, the list could be jumbled in a pseudorandom fashion by a code known to both the input and output processor.

We now want to analyze the amount of protocol start information/message and the average message delay for this protocol. In the remainder of the section, we assume that $\bar{x}$ is the length of the message in information bits and any protocol information associated with the message is not included in $\bar{x}$. For brevity in the following discussion, we only consider the case $T \geq \bar{x}$. For $T \geq \bar{x}$, the calculation of the amount of protocol start/information message proceeds as in [1]. Consider the entropy of the contents of a queue when it is sampled. On the assumption that the message arrivals form a Poisson process of rate λ, the probability that n messages arrive in T seconds is given by the Poisson distribution

$$P_{\lambda T}(k) = \frac{e^{-\lambda T} (\lambda T)^k}{k!}, \quad k \geq 0.$$

Thus the entropy of the number of message arrivals in T seconds is

$$H(P_{\lambda T}) = \sum_{k=o}^{\infty} \frac{e^{-\lambda T}(\lambda T)^k}{k!} \log_2 \frac{k! e^{\lambda T}}{(\lambda T)^k} \text{ bits.}$$

On the average, the number of messages in T seconds is λT, so that the average number of protocol bits/message transmitted about message arrival time is

$$\frac{H(P_{\lambda T})}{\lambda T}. \tag{7}$$

We now want to find the average message delay for this protocol. First we make an assumption which greatly simplifies the analysis. We assume that for $T \geq \bar{x}$, T is an integral multiple of $\bar{x}$, and that each frame can hold $T/\bar{x}$ messages. In effect then, we assume that protocol information at the start of a frame causes no message delay and we temporarily neglect the effect of improbably long protocol codes at the start of a frame which cause message queueing delays and less than $T/\bar{x}$ messages to be sent in a frame. We will show later that this assumption does not affect the validity of the results. For $T=\bar{x}$, the average message delay is given by

$$d = \left(1 + \frac{\rho^2}{2(1-\rho)}\right)\bar{x},$$

a well known result from buffer design [12], so that

$$\lambda d = \left(1 + \frac{\rho^2}{2(1-\rho)}\right)\frac{\rho}{n}. \tag{8}$$

The determination of the average message delay for $T>\bar{x}$ is more difficult because queues of this type are exceedingly difficult to handly analytically. For a discussion of some analytical results concerning queueing systems of this type, see [2], [4], [5], [8]. Invariably, almost all analytical results are given in transform form containing an expression which involves the determination of the roots of a transcendental equation. For a Poisson arrival process, these roots can be given as an infinite summation, which then requires numerical techniques to evaluate. Note that our queueing system is similar to the queueing system commonly referred to as the transportation problem in which a bus of fixed capacity arrives to a stop at fixed time intervals to pick up passengers arriving according to a Poisson process of rate $n\lambda$. The difference lies in the fact that in the message protocol, messages serviced by a frame must wait a certain fraction of the frame time to be serviced. Nevertheless, the number of customers waiting when the bus arrives is the same as the number of messages waiting at the start of a frame. From the distribution of this quantity, which is known in transform form, we may calculate $\bar{N}$, the average number of messages present in the system, and then using Little's result we may obtain the average message delay.

Let the random variable W be the number of messages waiting at the start of a frame in equilibrium. It is clear that the distribution of W is a function only of $T/\bar{x}$, which we call b, and the average message arrival rate per frame of the Poisson process, ρb. The generating function of W is given by [4]

$$W^*(z) = \frac{b(1-\rho)\,(z-1)\,P(z)}{z^b - P(z)} \prod_{t=2}^{b} \frac{z-\theta_t}{1-\theta_t} \qquad (9)$$

where P(z) is the generating function of the Poisson arrival process,

$$P(z) = \exp[\rho b(z-1)],$$

and

$$\theta_t = \lim_{w\to 1^-} z_t(w), \quad 2 \le t \le b,$$

where $z_t(w)$ is one of the b roots of the equation

$$z^b - wP(z) = 0$$

for w real and $0 < w < 1$. For our P(z) of interest, $z_t(w)$ is known to be given by an infinite summation ([4], [9]) which can be evaluated numerically.

We may now calculate the average message delay. Let V be the number of messages that arrive in T seconds. It is clear that

$$\bar{N} = E(W) + E(W-1)^+ + E(W-2)^+ + \ldots E(W-b)^+ + E(V)/2.$$

But

$$E(V) = \rho b,$$

$$E(W) = \left.\frac{dW^*(z)}{dz}\right|_{z=1},$$

and $E(W-j)^+$ may be determined from E(W) and $P\{W=k\}$, $k=0,1,\ldots,j$. These last quantities may be determined by differentiating the generating function $W^*(z)$. Finally, using Little's result we have

$$d^* = \frac{N}{\rho} - 1$$

where d^* is the average message delay given in units of $\bar{x}$. As a rough estimate of the results of these calculations, we find in following numerical studies that d^* is on the order of b. The average message delay d, in seconds, is

$$d = \bar{x}d^*$$

which gives

$$\lambda d = \lambda\bar{x}d^* = \frac{\rho d^*}{n} \qquad (10)$$

As one expects, (10) shows that λd is not a function of $\bar{x}$.

We can find an approximation to (10) for $b \gg 1$ as follows. For $b \gg 1$, we expect by law of large numbers type arguments such as used for example in the diffusion approximation to queueing behavior that

$$E(W) = b$$

and

$$d^* = \frac{b}{2} + \frac{\rho b}{2}.$$

This gives

$$\lambda d = \lambda \bar{x} d^* = \frac{\lambda T}{2} + \frac{\rho \lambda T}{2}. \tag{11}$$

We now want to find the limiting behavior of our protocol as $\rho \to 0$ with n, λ, and T fixed. As $\rho \to 0$, $\bar{x} \to 0$, and thus $b \to \infty$; using (11) we have

$$\lambda d \to \frac{\lambda T}{2}. \tag{12}$$

Since λT is fixed, the number of protocol bits/message remains

$$\frac{H(P_{\lambda T})}{\lambda T}. \tag{13}$$

Note that as $\rho \to 0$, the protocol performance given in (12) and (13) reduces to the performance of Gallager's first protocol, designed for networks without queueing delay.

We see from expressions (7)-(10) that the average message delay and average number of protocol bits/message depend on the quantities n, ρ, and the product λT. In Figure 4, we show the theoretical performance of this protocol for $n=100$; $\rho=.2,.5$, and $.7$; and a range of values of λT. Using (12) and (13), the limiting behavior as $\rho \to 0$ is also shown as a dashed line. To obtain the protocol performance for other values of n, note that the basic character of the protocol performance only depends on the product $n\lambda$, since this is the total arrival rate to the queue. It is clear from $n\lambda\bar{x}=\rho$ that, for example, when n increases by a factor of 10, λ must decrease by a factor of 10 if ρ and $\bar{x}$ do not change. This results in a shift of the curve along the horizontal λd axis by a factor of 10. Note from Figure 4 that as λd becomes small there is a sharp increase in the number of protocol bits/message for each value of ρ. This is because the number of idle bits/message increases without bound; the same effect occurred in the pure addressing protocol.

We now want to consider two specific implementations of this protocol. The first implementation is optimal. For a particular source, we want to encode the random variable X which tells how many messages have arrived in the previous frame of length T (assume $T \geq x$). Define the random variable Y by

$$Y = \begin{cases} 1, & X > 0 \\ 0, & X = 0. \end{cases}$$

It is clear that

$$H(X) = H(XY) = H(Y) + H(X|Y). \tag{14}$$

We encode the random variable X by using two separate codes; the first sends Y and the second sends X given Y. If each code is optimal, then it follows from (14) that we have constructed a protocol using

$$H(X) = H(P_{\lambda T})$$

bits per source per frame. We encode the random variable Y by using a run-length code over the sources which indicates whether a source has at least one message waiting to be transmitted; the run-length code is known to be optimal for the memoryless binary source [6]. We then follow the run-length code with an optimal variable length code giving the number of messages waiting at each source queue that is not empty; this encodes $H(X|Y)$. It follows that the number of protocol bits/message for this scheme is given by

$$\frac{H(X)}{\lambda T} = \frac{H(P_{\lambda T})}{\lambda T}$$

which is (7).

The second implementation is more characteristic of data networks. In this scheme, we convey protocol information about message arrivals in the previous frame by first sending the total number of messages that arrived using a variable length code and then sending the $\lceil \log_2 n \rceil$ bit addresses for the messages that arrived. The number of protocol bits/message for this scheme is

$$\frac{H(P_{n\lambda T})}{n\lambda T} + \lceil \log_2 n \rceil$$

assuming $T \geq \bar{x}$. A comparison of the two implementations shows that the addressing scheme is very inefficient as b becomes large, if n is not small.

As mentioned at the beginning of this discussion, the message delay analysis has neglected the effect of improbably long protocol codes at the start of a frame. To verify that this has a negligible affect on protocol performance, the two implementations were simulated for $n=100$, $\rho=.5$, $\bar{x}=80$, and a range of values of T. The simulation and theoretical results show close agreement; again, for brevity, the simulation results are not presented.

The preceeding protocol used past information of the message arrivals at previous frames so that at each frame it only needed to encode the conditional information of message arrivals in the preceeding frame. Using the time frame approach of the previous protocol, a protocol can be constructed which does not encode conditional information and therefore does not require the input and output processor to keep a list of messages in the queue. Simulation and theoretical results for this protocol agree closely with the preceeding protocol. Another protocol which was simulated was an asynchronous loop scheme. Simulation results for this protocol showed similar behavior to the two previous protocols.

ACKNOWLEDGEMENT

This work was supported in part by NSF Grant ENG79-08073.

References

1. R.G. Gallager, "Basic limits on protocol information in data communication networks," IEEE Trans. on Info. Theory, Vol. IT-22, pp. 385-398, July 1976.

2. J.W. Cohen, The Single Server Queue, Wiley Interscience, New York, 1969.

3. L. Kleinrock, Queueing Systems, Vol, 1, 1975; Vol. 2, 1976, Wiley Interscience, New York.

4. A.G. Konheim and B. Meister, "Service in a loop system," J. of the Ass. for Computing Machinery, Vol. 19, No. 1, pp. 92-108, January 1972.

5. N.T. Bailey, "On queueing processes with bulk service," J. Royal Statistical Society, Vol B 16, pp. 80-87, 1955.

6. R.G. Gallager, "Optimal source codes for geometrically distributed integer alphabets," IEEE Trans. on Info. Theory, Vol. IT-21, pp. 228-230, March 1975.

7. A.G. Konheim and B. Meister, "Waiting lines and times in a system with polling," J. of the Ass. for Computing Machinery, Vol, 21, No. 3, July 1974, pp. 470-490.

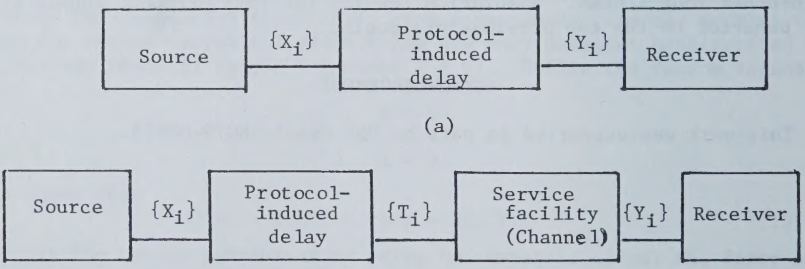
8. P.E. Boudreau, J.S. Griffin, and Mark Kac, "An elementary queueing problem," Am. Math. Monthly 69, 8 (1962)., pp. 713,724.

9. E.M. Wright, "Solution of the equation $ze^z=a$," Bull. Am. Math Monthly 65 (1959), pp. 89-93.

10. E. Reich, "Waiting times when queues are in tandem," Ann. Math Statistics 28, pp. 768-773, 1957.

11. W. Feller, An Introduction to Probability Theory and its Applications, Vol. 2, 1971, John Wiley, New York.

12. M. Schwartz, Computer-Communication Network Design and Analysis, Prentice-Hall, Englewood Cliffs, NJ 1977.



(a)

(b)

Figure 1.

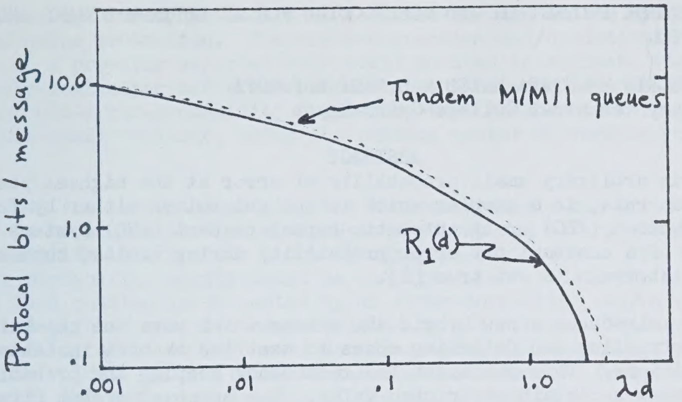
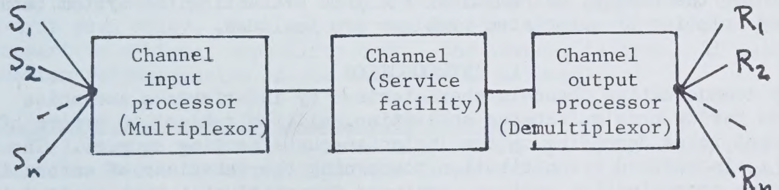


Figure 2.



S₁
S₂
Sₙ
Channel input processor (Multiplexor)
Channel (Service facility)
Channel output processor (Demultiplexor)
R₁
R₂
Rₙ

Figure 3.

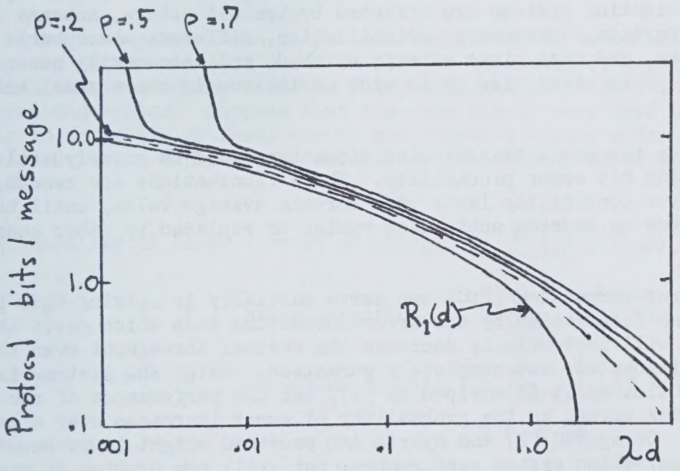


Figure 4. n = 100.

QUANTIZED NOISE PROTECTION AND MULTICODING FOR AN OPTIMUM BINARY ARQ COMMUNICATION

MOHI-ELDIN MOUSTAFA RATEB & M.NAGI ELYOUSFI
Military Technical College-Cairo-Egypt

ABSTRACT

Attaining an arbitrary small probability of error at the highest possible transmission rate, is a problem which is not yet solved either by forward-error-correction (FEC) or by automatic-repeat-request (ARQ) systems. The suppostion of a constant bit error probability during limited communication time intervals is not true [1].

This paper introduces a new hybrid ARQ system which uses the capabilities of error-correcting and detecting codes to meet the channel instantaneous noisy conditions. This minimizes the redundancy keeping the probability of error below a certain restricted value. The proposed system fits more closely the high signalling rate channels, and its efficiency increases by rate increase. The idea serves efficiently in building future error-free communication systems with maximum possible throughput. All considerations concerning the design, mathematical analysis evaluating the system behaviour and solution of associated problems are included.

1. INTRODUCTION

Binary communication channels characterised by interferance and noise sources having nonsimultaneous activities, will be subject to errors of different rates depending on the instantaneous affecting sources. The work [1] introduced a contribution concerning the behaviour of errors in telephone communication systems, achieved the conclusion that error data are widely varied and depending on the average error rate in evaluating the channel behaviour, is completely misleading. On the other hand, satellite communication systems are affected by transit outage, antenna side lobe interferance, ionospheric scintillation, different atmospheric attenuation means, and many other effects which do not necessarily occur simultaneously. This gives rise up to wide variations in the channel bit error rate.

In all cases the whole transmission situation could be closely evaluated as a fluctuating bit error probability. Those fluctuations are random, and the bit error probability lasts on a certain average value, until the dominating source or sources activities vanish or replaced by other sources activities.

Forward-error-correction (FEC) can serve partially in solving this problem. Encoding the information by the error-correcting code which meets the worst conditions will subtentially decrease the channel throughput even though those conditions are not completely guranteed. Using ARQ systems is a better solution as it is advised by [1], but the performance of such channels goes worse, as the probability of error increases over a certain threshold. Using MRQ [2] and hybrid ARQ provided slight improvement in total communication system performance, but still the problem of maximizing the system throughput for a restricted small bit error probability, to meet the mentioned above wide variations in practical channel noisy conditions, is not solved.

2. QUANTIZED NOISE PROTECTION

2.1 Basic Idea

For a given permissible probability of erroneous decoding, the highest possible average throughput could be achieved through an ARQ system using quantized noise protection. The error-correction and/or detection capabilities of a properly selected code could be used to estimate the instantaneously dominating bit error probability. This requires an efficient sufficient bit-error-probability estimator, that reaches decision, with a permissible small variance, using the minimum number of received code blocks.

Suppose that the estimator exists, the maximum probable bit-error-probability is known via careful channel statistics, and a very small allowable probability of erroneous block decoding is restricted. The total bit error probability margin could be quantized into a definite number of quanta. Each quantum is protected by an error-correcting and/or detecting code of the highest rate and capability among all others. This code keeps the probability of block erroneous decoding within the quantum, below the permissible value. On the basis of information extracted from received code blocks, the estimator decides if the noisy conditions are within the quantum, higher or lower. If higher an instruction is transmitted to a controllable-rate encoder in the sending terminal through the power unlimited feed back channel, and simultaneously to a controllable-rate decoder in the receiving terminal with a time delay, to compensate the round-trip path delay. Both encoder and decoder will switch over at the same moment to a higher capability code. The inverse is true. If within the quantum retransmission of the same code is allowed.

2.2 Criteria for Quantum-Protecting Code Selection

In the following discussion, it is assumed an ARQ system in which retransmission of a detectable-error code word is permitted, till a correctable (or probably a nondetectable) code word is received. Using MRQ [2] will improve the system performance, but this does not contradict with the analysis introduced in the following pages.

At first we have to judge for a block code consisting of k-bit message blocks encoded into n-bit code words to form an (n,k) linear block code, how the available redundancy could be fully utilized in conjunction with our proposed ARQ system. Suppose that the code blocks used have the capability to correct c or fewer errors and detect d errors where d > c. Assume a B.S.C having a bit-error-probability p and independant errors, it holds that:

$$P_c \text{ (Probability of block correct decoding)} = \sum_{i=o}^{c} \binom{n}{i} p^i (1-p)^{n-i} \qquad (1)$$

$$P_d \text{ (" " " error detection)} = \sum_{i=c+1}^{d} \binom{n}{i} p^i (1-p)^{n-i} \qquad (2)$$

$$P_n \text{ (" " " erroneous decoding)} = \sum_{i=d+1}^{d} \binom{n}{i} p^i (1-p)^{n-i}$$

$$= (1-P_c-P_d) \qquad (3)$$

Consider the previously mentioned ARQ system and assuming ideal conditions, then it holds that

$$P_{cs} = P_c/(1-P_d) \text{ and } P_{ns} = P_n/(1-P_d) = 1 - P_{cs} \qquad (4.a,b)$$

and the system throughput

$$T = k(1-P_d)/n \qquad (5)$$

For MRQ systems this represents the lower bound and we can expect better results. Now let us introduce the following definition which characterises the efficiency of quantized noise protection ARQ system:
Definition: The figure of merit of a quantized ARQ system is the $(P_{cs}.T)$ product.
The code which maximizes this figure is the best for quantized ARQ systems.

$$F = P_{cs}.T = (k.P_c)/n \qquad (6)$$

This means that the code used should have the maximum correction capabilities in addition to some detection capabilities. As an example, let the code (8,4) be of interest. We have the possibility either to encode by the Hamming code $d_{min}=4$ capable to correct one and detect two random errors, or to have a code capable to detect all three or fewer errors. Let $p = 0.05$ which is very high w.r.t normal situations.
Then for the Hamming code $P_{ns} = 0.056$, $T = 0.4969$
and for the detection code $P_{ns} = 0.0006$, $T = 0.3319$
Repeating the same procedure with the code (16,11) yields for the Hamming $P_{ns} = 0.05$, $T = 0.587$ and for the detection code $P_{ns} = 0.04$, $T = 0.314$.
It is observed that as the block code length increases the difference in the probability of erroneous decoding decreases while the throughput difference greatly increases. Since in practical communication systems long code blocks are used, this verifies the basic criterion for selection in general.

In quantized ARQ systems the problem of higher probability of erroneous decoding is easily solvable. It leads to a decrease in the upper limit margin of the quantum which the code protects in the original system design.

-The second criterion for a code selected to guard (protect) the i^th^ quantum is to satisfy the following inequalities

$$P_{ns}(P_i) \leqslant P_{np} \quad \text{or} \quad P_{cs}(P_i) \geqslant P_{cp} \qquad (7)$$

where

P_i = the upper limit margin of bit-error-prob. of the i^{th} quantum

P_{np} = the restricted probability of erroneous decoding of a code block.

3. QUANTIZED ARQ SYSTEM BASIC CONSTRUCTION

3.1 System Basic Construction

For simplicity purpose, the discussion will be restricted to a binary point-to-point system. Due to additional merits provided by ARQ systems with memory (MRQ) we build this system as a quantized ARQ system with memory (QMRQ). The simplified system construction is shown in Fig. 1. Looking forward for attaining the best system properties we used the continuous ARQ technique which retransmits the only code block with detected error.

Information is supplied from data source (DS) to controllable-rate encoder (CRE). The information code blocks are encoded into (n_i,k) block codes according to the instantaneously operating q_i quantum . The code selection is governed by the feedback control block (FBC). This block receives instructions from the estimator (E) in the receiving terminal via feedback power unlimited channel. The encoded blocks are stored in the main buffer (MB) which allows the information to the output via the modulator (M), and simultaneously supplies this information to a temporary buffer (TB), during its exit. When (FBC) receives an instruction for repetition, the switch will be in down position and the information is extracted from the (TB).

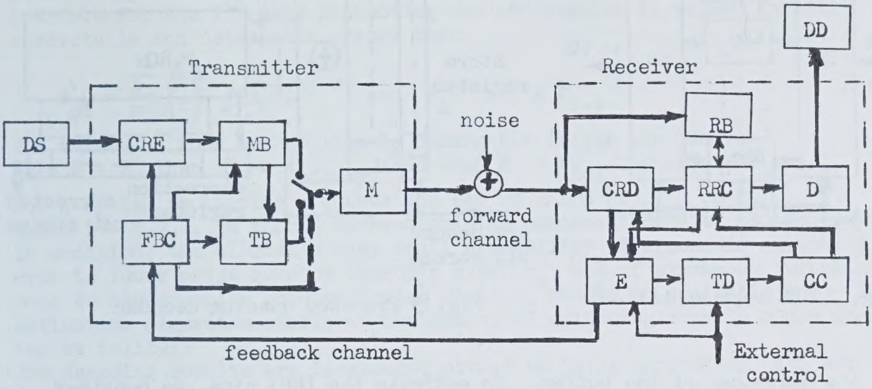


Fig. 1 Quantized ARQ with memory (QMRQ)

In the transmitter side, the estimator (E) in fact controls the operation of the transmitter and the entire system. It receives information from the controllable-rate decoder. This information is the succession of the decoding results. On the basis of this information the estimator decides either to remain in the operating quantum and retransmission of error detectable blocks is allowed, or to switch over to one of the two adjacent quanta. The instructions are applied directly through the feedback channel to the (FBC) block in the transmitter, and to time delay block (TD) in the receiver. The (TD) delays these instructions by a total round trip path delay. The delayed instructions are applied to the central control block (CC). The (CC) block execute those instructions by leading the arriving code blocks into the corresponding decoder, and to acknowledge the reception redundancy error correction (RRC) for changing the processing algorithm to adapt the new conditions.The final decoding result is applied from the (RRC) to the data destination (DD) via the demodulator (D). The redundancy buffer (RB) operates in conjunction with (RRC) for fulfilling the MRQ algorithm [2]. In case of systems like satellite communication systems the round trip path delay may change due to drifts of satellite out of its orbit. Information from the ground keeping orbit station should be provided to control the round trip path delay according to true situation.

3.2 Design of Individual System Blocks

3.2.1 Transmitter Controllable-Rate-Encoder: The general representation of this encoder is in Fig. 2. The input information blocks are applied from the data destination to a demultiplexer (DE) to the corresponding quantum-protecting encoder. The output encoded blocks are applied to the main store via a multiplexer (MU). Both (DE) and (MU) are controlled by the feed back control (FBC). A general controllable-rate "block" code encoder can replace this device if it exists. In Fig. 2,m quanta are assumed.

3.2.2 Transmitter Buffers: The (MB) is a RAM which matchs the variable rate input encoded information, with the output information flow demanded by the channel conditions. It always supplies the tempory memory by the output code blocks at its exit and to the output. When a specific block retransmission is requested the (MB) stops reading out information while the input encoded information continues being written into. The encoded blocks are labled to identify the code used, to determine the code length

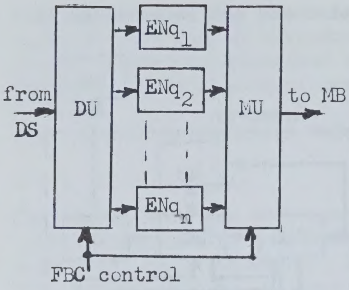


Fig. 2 CRE representation

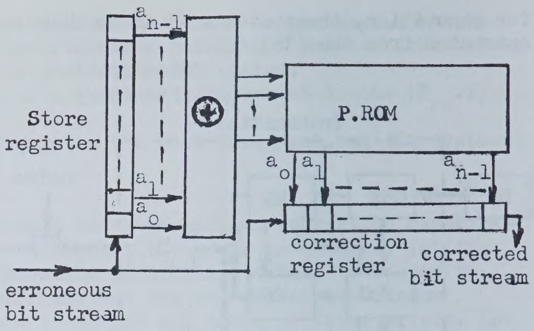


Fig. 3 Proposed quantum decoder

at its exist out of the buffer. To estimate the (MB) size, we consider that all code words during an overall trip path delay are requested for retransmission, the maximum allowable number of requests. In this case the memory size will be:

$M = N.R_{max}$ (bits) execluding additional synchronizing and signalling bits
where N = number of bits transmitted during a round trip path delay
R_{max} = maximum number of allowable retransmission.

However careful statistics can reduce this size to a large extent. The memory size is seen to be proportional in a close sence to rge signalling rate. Code segments are not allowed. The switch is in down position only after transmission of the total code block.

The temporary buffer (TB) is devoted for storing the encoded blocks for several retransmissions, until R_{max} cycles are exhausted. The (FBC) block assign the address for each code word located in (T.B). A simple method for labeling the stored code words is the time labeling. The (FBC) block lables the address of code blocks stored in (T.B) according to the time order of instructions arrival. After a round trip time delay, if a repeat request is detected the memory location where this code word is stored is exactly determined, and the contents are read out and newly labled. However the labeling cycle is finished after a time which is equal to the round trip delay times the maximum number of allowable retransmissions. The labled code words which receives no repeat request after a round trip time delay, are discarded and replaced by new code words. This minimizes the memory size to a large extent.

3.2.3 The Receiver Controllable rate decoder (CRD): The general representation is the same as the (CRE) with quantum-encoders replaced by decoders. Since the codes used may not essentially have cyclic properties, we propose the decoder in Fig. 3 representing one of the quantum-decoder blocks. The resultant syndrome generated by parity-check equations is applied to a P.R.O.M which instantaneously corrects the corresponding error pattern in the shift register temporary store. The PROM and shift register could be manufactured in one I.C chip overcoming the problem of I.C pins.

3.2.4 The Estimator: It estimates on the basis of information extracted from erroneously received code words (correctable or detectable), if the noisy conditions represented by the bit error probability are within the operating quantum or outside (higher or lower). The bit error probability margins could be replaced by the protecting code correction plus detection probability. The function is monotonic increasing and the replacement is possible. For the $i^{\underline{th}}$ code protecting the $i^{\underline{th}}$ quantum is valied for all correctable and detectable errors that

$$P'_{di} = \sum_{r=1}^{d} \binom{n_i}{r} P(1-P)^{n_i-r} \quad \text{and} \quad P_i \geqslant P \geqslant P_{i-1} \tag{8}$$

The values of this probability is remarkable inside the quantum.
When $P = P_i$ then $P'_{di} = P'_{di(max)}$ and when $P = P_{i-1}$ then $P'_{di} = P'_{di(min)}$
For a system protecting m quanta the set of pairs $[P'_{di(min)}, P'_{di(max)}]$ where $i = 1,2,...,m$ should be available as permanent data to the estimator. In addition, the allowable very small probability of erroneous switching over to lower noise quantum (error's risk) r_e, and of erroneous switching over to higher noise quantum (rate's risk) r_r should be available too. The estimation algorithm starts at the moment of switching over to a new quantum as follows:
The decoding results are introduced either as being correct or with error (correctable or detectable). Let D_i be the number of code words with detectable error among J tested code words, then [8,9]:

$$L_J = \frac{\log[r_e/(1-r_r)] + J \log[(1-P'_{di(min)})/(1-P'_{di(max)})]}{\log[P'_{di(max)}/P'_{di(min)}] - \log[(1-P'_{di(max)})/(1-P'_{di(min)})]} \tag{9}$$

and

$$H_J = \frac{\log[(1-r_e)/r_r] + J \log[(1-P'_{di(min)})/(1-P'_{di(max)})]}{\log[P'_{di(max)}/P'_{di(min)}] - \log[(1-P'_{di(max)})/(1-P'_{di(min)})]} \tag{10}$$

which reduces to the solution of two linear equations in the form
$L_J = K_1 + K_2 J$ and $H_J = K_3 + K_4 J$

If L_J is not an integer it is replaced by the largest integer less than L_J, and if H_J is not an integer it is replaced by the smallest integer less than H_J. In each step the estimator receives a code word result it tests if:

$$L_J \geqslant D_i \geqslant H_J \tag{11}$$

If $D_i \geqslant H_J$ a decision to switch over to higher noise quantum is declared.
" $D_i \leqslant L_J$ " " " " " " lower " " " "
If within both margins operating with the same code is decided, and retransmission of detectable-error code words (noncorrectable) is allowed. The proposed sequential estimator can solve efficiently this problem. It could be simply realized by microprocessor technique.

4. EVALUATION OF SYSTEM PERFORMANCE AND PARAMETERS

Referring to criteria introduced, and aiming to achieve a general mathematical representation, we shall assume that the first noise quantum where the probability of bit error is minimum is protected by a single parity-check code, the second by a code capable to correct one error and detect two, the third by a code capable to correct two and detect three, an so on. In general the code protecting the $i^{\underline{th}}$ quantum will have the correction capabilities of i-1 or fewer errors and detection capabilities of i errors. We shall denote the upper level margin of the bit error probability of the $i^{\underline{th}}(q_i)$ by P_i. This margin is the value of P_i at which the overall system probability of block incorrect decoding P_{nsi} is equal to the restricted permissible erroneous decoding probability P_{np}. The proposed quanta protection is not the only solution. We may for example protect the first quantum by single parity code, the second by a double-error detection, the third by one error correction and double error detection, and so forth. This allows lower quanta widths and better throughput, but this depends on the estimator capabilities in final decision.

4.1 Calculation of the Bit error Probability Margins

At the upper margin of the protected quantum q_i the bit error probability P_i must secure a system probability of error $P^i_{nsi} < P_{np}$ for the quantum protecting code. Assuming unlimited retransmission cycles, then at the upper limit threshold of the i^{th} quantum is valied:

$$P_{nsi(max)} = P_{ni(max)}/(1-P_{di(max)}) = P_{np} \tag{12}$$

also an overall system probability of correct decoding

$$P_{csi(min)} = P_{ci(min)}/(1-P_{di(max)}) = 1-P_{np} \tag{13}$$

eleminating $(1-P_{di(max)})$ from (12), (13) yields

$$P_{ni(max)}/P_{ci(min)} = P_{np}/(1-P_{np}) = D \tag{14}$$

For a selected (n_i,k) code correcting (i-1) or fewer random errors and detecting i errors to protect the quantum q_i is valid

$$D = \sum_{r=i+1}^{n_i} \binom{n_i}{r} P_i^r(1-P_i)^{n_i-r} / \sum_{r=o}^{i-1} \binom{n_i}{r} P_i^r(1-P_i)^{n_i-r} \tag{15}$$

The rearrangement of this equation yields

$$D = \sum_{j=i+1}^{n_i} \binom{n_i}{j} X_i^J / \sum_{j=o}^{i-1} \binom{n_i}{j} X_i^j \tag{16}$$

where

$$X_i = P_i/(1-P_i) \quad \text{and} \quad D = P_{np}/(1-P_{np}) \tag{17}$$

The solution of the equations of type (15) yields only one real root and all others complex. This real root determines the threshold value P_i for such code. Since the probability of occurrence of errors of order higher than i+1 is negligible then (16) will reduce to

$$D \doteq \binom{n_i}{i+1} X_i^{i+1} / \sum_{j=o}^{i-1} \binom{n_i}{j} X_i^j \tag{18}$$

As example for a code (n_2,k) protecting the second quantum it holds that:

$$D \doteq \binom{n_2}{3} X_2^3/\left[\binom{n_2}{1} X_2 + 1\right] \quad \text{or} \quad X_2^3 = K/\left[\binom{n_2}{1} X_2 + 1\right]$$

where $K = D/\binom{n_2}{3}$ and $X_2 = P_2/(1-P_2)$. We solve graphically for X_2 and then obtain $P_2 = X_2/(1+X_2)$

4.2 Evaluation of System Throughput and Probability of Error

4.2.1 Throughput: For restricted very small P_{np} and long enough codes, the quantum width will be essentially small. The distribution of bit error rate within the quantum will be most likely uniform. The bit error probability now is considered within the quantum as a continuous random varible having a uniform density. For an ARQ system the throughput is given by:

$$T = k/[n(1-P_d)]$$

which represents the lower bound of the MRQ throughput. For a quantized protection ARQ system, the same relation is valid within the quantum. For the i^{th} quantum it holds:

$$T_i(P) = \frac{k}{n_i}\left[1-P_{di}\right] = \frac{k}{n_i}\left[1-\binom{n_i}{i} P^i(1-P)^{n_i-i}\right] \tag{19}$$

For correct evaluation of the system throughout we introduce T_i (the average throughput of the i^{th} quantum) as:

$$\bar{T}_i = \int_{P_{i-1}}^{P_i} F_i(P)\, dP \tag{20}$$

$F_i(P)$ is the probability density function of P within q_i which is uniform hence

$$\bar{T}_i = \frac{K}{n_i(P_i-P_{i-1})} \int_{P_{i-1}}^{P_i} \left[1-\binom{n_i}{i} P^i(1-P)^{n_i-i}\right] dP \tag{21}$$

Expanding and integrating term by term yields:

$$\bar{T}_i = \frac{k}{n_i}\left[1-\binom{n_i}{i}(P_i^i-P_{i-1}^i)\sum_{J=0}^{n_i-i}(-1)^J\binom{n_i-i}{J}(P_i^J-P_{i-1}^J)/(J+i+1)\right] \tag{22}$$

Neglecting higher order terms leads to a very rough evaluation of T_i

$$\bar{T}_i > \frac{k}{n_i}\left[1-\binom{n_i}{i}(P_i^i-P_{i-1}^i)/(i+1)\right] \tag{23}$$

This indicates that as the quantum width decreases, in addition to the corresponding decrease of probability of block error nondetection, the average quantum throughput increases to reach in the limit case the FEC throughput. This calls for greater number of quanta and highly efficient estimators. If by careful statistics, the function $P(q_i)$ (which assigns to each quantum its probability of occurrence) is known, and for m operating quanta is valid that

$$\sum_{i=1}^{m} P(q_i) = 1 \tag{24}$$

$$\text{then } \bar{T} = \sum_{i=1}^{m} \bar{T}_i\, P(q_i) \tag{25}$$

Where $\bar{T}$ is the overall system average throughput.

4.2.2 Probability of Error: Attempting a general expression for evaluating the average block nondetectable error/quantum yields a very complicated formula. But, we can evaluate the upper and lower bounds of this probability.

$$P_{np} \geqslant P_{nsi} \geqslant P_n(P_{i-1})/[1-P_d(P_{i-1})] \tag{26}$$

where $P_n(P_{i-1})$ and $P_d(P_{i-1})$ are the nondetecting and detecting code block probabilities evaluated at the lower quantum margin P_{i-1}. For an enough narrow quantum, P_n could be approximated by a straight line segment and a rough estimation of this average yields

$$\bar{P}_{nsi} = \left[P_n(P_{i-1}) + P_{np}\{1-P_{di}(P_{i-1})\}\right]/2\left[1-P_d(P_{i-1})\right] \tag{27}$$

Fig. 4 provides a simple graphical interpretation of an ARQ system based on quantized noise protection, transmitting six information bits. If statistics confirm that $P > 10^{-2}$ occurs with zero probability, and a permissible probability of block error nondetection is restricted by $P_{np} < 10^{-4}$ or $P_{cp} > 0.9999$. A proposed quantized system, guards the first quantum

(q_1) by the parity-check code (7,6), (q_2) by the Hamming (11,6) code capable to correct one error and detect two, and (q_3) by the half-rate convolutional code capable to correct two errors without allowable retransmission. If a distribution of bit error fluctuations yielded $P(q_1) = o.6$, $P(q_2) = o.3$ and $P(q_3) = o.1$. Calculation leads to an average system throughput T = o.7271, which is much higher than o.5 if convolutional code is only used.

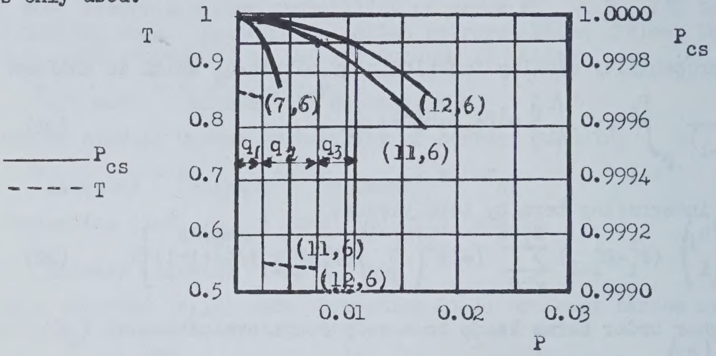


Fig. 4 Graphical interpretation

5. CONCLUSION

Quantized MRQ (QMRQ) system is the ultimate implementation of FEC and MRQ merits. It is the only direction which guides to an error-free and a highest throughput comm. systems. In the near future is expected higher signalling rate comm. systems, and more development in technology and science. The higher signalling rate provides the estimator with the neccessary information in less time, the technology solves the problem of greater buffers to meet the higher rate and science offers new controllable-rate codes that maximizes the number of quanta and thus approaching an ideal quantized MRQ system. Thus the QMRQ is the system of future utilization.

Still the known coding techniques envolve a remarkable redundancy losses. Error-correcting and/or detecting codes, of constant length, which inside their structure have various capabilities fitting a certain alphabet source distribution, will minimize the code length. Composite codes [10] solve this problem. Wonderful results are expected when such codes are used with a QMRQ system.

We have restricted our analysis to random errors to attain specific results for the system evaluation. However if the channel statistics envolve more burst occurrence, quantization is also applicable with burst error correcting and detecting codes and similar results are achieved.

6. REFERENCES

(1) H.O. Burton, and D.D. Sullivan, "Errors and Error Control" Proceedings of the IEEE. Vol. 6o, No. 11, November 1972.

(2) Fradeeps. Sindhu, "Retransmission Error Control with Memory", IEEE Trans on Comm., Vol. Com. 25, No. 5, May 1977.

(3) W.W. Peterson and E.J. Weldon, Jr, "Error Correcting Codes", 2nd edition, MIT Press, Cambridge, Massachusetts, 1970.

(4) J.I. Metzner and K.C. Morgan "Coded Binary Decision Feedback Communication Systems", IRE Trans. Comm. Sys. Vol. CS.8, pp. lol - 113, June 1960.

(5) G.D. Forney, Jr, "Coding and its Application in Space Communications", IEEE Spec rum, Vol. 7, pp. 47 - 62, June 1970

(6) Sastry, A.R.K., Laveen No., "Hybrid Error Control using Retransmission and Generalized Burst Trapping Codes", IEEE Trans on Comm. 24 , No. 4, 385 - 393, June 1976.

(7) Robert G. Gallager, "Information Theory and Reliable Communication", Massachusetts Institute of Technology, 1968.

(8) I. Miller and J.E. Freund, "Probability and Statistics for Engineers" Prentice-Hall, Inc., June 1965.

(9) A. Wald, "Sequential Analysis", John Wiley, 1947.

(10) M. Moustafa Rateb, M. Elyousfi, "Maximum Rate Minimum Error Composite Codes", Septieme Colloque sur le Traitement du Signal et ses Applications, Gretsi, 34/1 - 34/7, May 1979.

ON THE PROPOSED FEDERAL STANDARD 1033 FOR TELECOMMUNICATIONS

EARL D. EYMAN
ELECTRICAL AND COMPUTER ENGINEERING
UNIVERSITY OF IOWA
IOWA CITY, IOWA 52242

ABSTRACT

To meet the growing need for the performance of communication systems from the point of view of services supplied to end users, a new Federal Standard is being developed. This standard is being applied to the specifications of a computer system in the control of a matrix of machines.

INTRODUCTION

Proposed Federal Standard 1033 specifies a set of parameters that can be used to describe the performance of any data communication system from the point of view of digital services supplied to end users. This report is to show the effect that the values of these proposed parameters have in defining communication requirements for the control of a matrix of machines in a factory with a general purpose computer. The parameters are to be system independent. The performance requirements for the system are best defined in terms of a relationship between the possible values of each parameter and the effectiveness of the user process; i.e., loss in dollars versus parameter values; this may be done with a set of curves.

In order to demonstrate the use of the parameters in evolving communication requirements, a hypothetical example in which the process is a company is being developed. The company has a factory which is a matrix of machines taking in raw materials and producing end products which have an increase in value. At present each machine has a person operating it, and the company has a general purpose computer in a remote building at some distance (a couple of miles) from the factory. This computer is presently performing various company functions such as payroll, inventory of finished goods, raw materials, purchase orders, employee records, accounting, word processing, and other general business functions.

A decision has been made to automate the factory using the general purpose computer in the process control mode. In addition a corporate ground rule requires (1) that the existing operating sytem in the computer is to remain unchanged, and (2) the existing hardware and control of the machines is to remain unchanged.

Other assumptions include a value for the raw material, a higher value for the end product, a maximum capacity that can be produced by assuming full utilization, any departure on the part of the communication system will reduce the profit by some amount, and a formula exists to describe the effectiveness of the communication system.

Factory Profit ($/unit time)

= {rate of production (parts/unit time)}·{average profit ($/part} minus overhead minus communication cost. We just subtract the communication cost as we would if the system was being rented. A negative sign will be assigned to ruined parts (use same averages). (Note: A decrease in efficiency refers to rate overhead: the computer system is a component.) Reliability relates to rate. We may also use zero instead of a negative sign for ruined parts.

The thought in specifying the communication system is that a process control program would be added to existing hardware as an applications program and would be linked to the other programs including inventory and orders in such a way that the program could decide what parts would be produced each day and direct the operation of the machines on line to produce the proper complement of parts. A supervisor will examine a computer readout each day before the computer takes over and runs the factory. The question being asked is what sort of communication system should be developed to interconnect the process program in the general purpose computer with the machine buffers.

DESCRIPTION OF MODEL

The description of the model will consist of the following five subjects:

(1) a definition of the User/System interfaces;
(2) definitions of the events for the parameters at the interfaces during transactions;
(3) an association of specific events with corresponding reference events; for instance, the specific event SYSTEM CALL will be associated with the reference event "start of access";
(4) a relationship between performance values and dollars will be defined; and
(5) two different systems with corresponding performance values will be postulated and compared from the point of view of their (a) intrinsic cost, and (b) effect on factory profit.

TRANSACTION PROFILE

The flow of information will be considered (a) from the process control program to the buffer of a particular machine, and then (b) from the terminal of a particular machine back to the process control program.

to the buffer of a particular machine, and then (b) from the terminal of a particular machine back to the process control program. Prior to the movement of information from the process control program (PCP) to the buffer of a particular machine, the PCP will examine orders, inventory, raw material, and related information, and decide what part is to be produced. Then the PCP will generate a file of data to direct the first machine operation to be performed in producing that part. The diagram shows the central computer and a machine connected by an unspecified means of transmission. The components are shown to the needed extent to permit the identification of the two user/system interfaces; one is between the operating system program and the process control program and the other is between the transmission facilities and the buffers of the machine (or the drop side of the terminal). The terminal and machine are connected to the computer as shown. The following seven steps constitute the transaction profile which will be referred to when the reference events at the user/system interfaces are used.

(1) Prior to the initiation of information transfer between the process control program (PCP) and the machines, the communication control program and the machine buffers will be readied to accept the associated signals issued during the communication process.

(2) The PCP will issue a request or SYSTEM CALL to the operating system program (OSP) to transfer the file of data from the buffer in the communication control program (CCP). The request identifies the machine destination address, the starting address of the data file, and the number of words in the file, ten for this example. This request will be queued with the OSP and sent sometime later to the CCP.

(3) At some later time, depending on the transmission capacity and other requests, the OSP will serve the request, SYSTEM CALL, by moving the data from the user file in a buffer in the PCP to a buffer in the CCP via a LOAD INSTRUCTION and then notifies the CCP of the availability of the data; this completes the transfer. The CCP will establish any necessary connections between the central computer and machines and output the information to the communication lines.

(4) When the entire file of data is in the machine buffer (the OSP will count the ten words and issue a FINISH signal), it goes to a comparator and typically generates a so-called "error signal" which controls the machine. This is data processing and is the end of the transaction.

(5) Going back, each time the machine completes an operation it generates a buffer of data which is stored in the control terminal. This data identifies the machine, part, past characteristics, and special status. This information is to be transferred to the PCP so that it can be used for the next step.

(6) The sequence here is that the machine fills the buffer and issues a transfer request or a REQUEST TO SEND. Then in the local mode the information is displayed. That same request is going out to some component in the system to request the reading of the information out of the buffer to the central computer.

(7) The system will then read the data from the buffer and issue a signal to the control terminal when the data has been read. The information is transferred to the CCP of the central computer. Then the CCP issues a request to the OSP. The OSP transfers the data to the PCP buffer and notifies the PCP that the data has been delivered and stored.

The drawings in Figs. 1, 2, and 3 show, respectively, the parameter development overview, the sample space for a typical function, and the access parameter definitions. Fig. 4 gives a summary of the selected performance parameters. Seven of the selected parameters were used to examine two designs of computer communication systems.

In summary, a computer controlled system can either increase or decrease profits depending on the relative overheads. Also some system even though more expensive can increase the profit so as to pay for the system plus an increased return on the original investment.

REFERENCES

1. Kleinrock (1976), Queueing Systems, Volume 2: Computer Applications (John Wiley and Sons, Inc., New York, NY).
2. Payne (1978), ARPANET host to host access and disengagement measurements Office of Telecommunications Report No. 78-U.S. Dept. of Commerce, Boulder, CO 80303.
3. Seitz et al (1978), Digital Communication Performance Parameters for Proposed Federal Standard 1033.

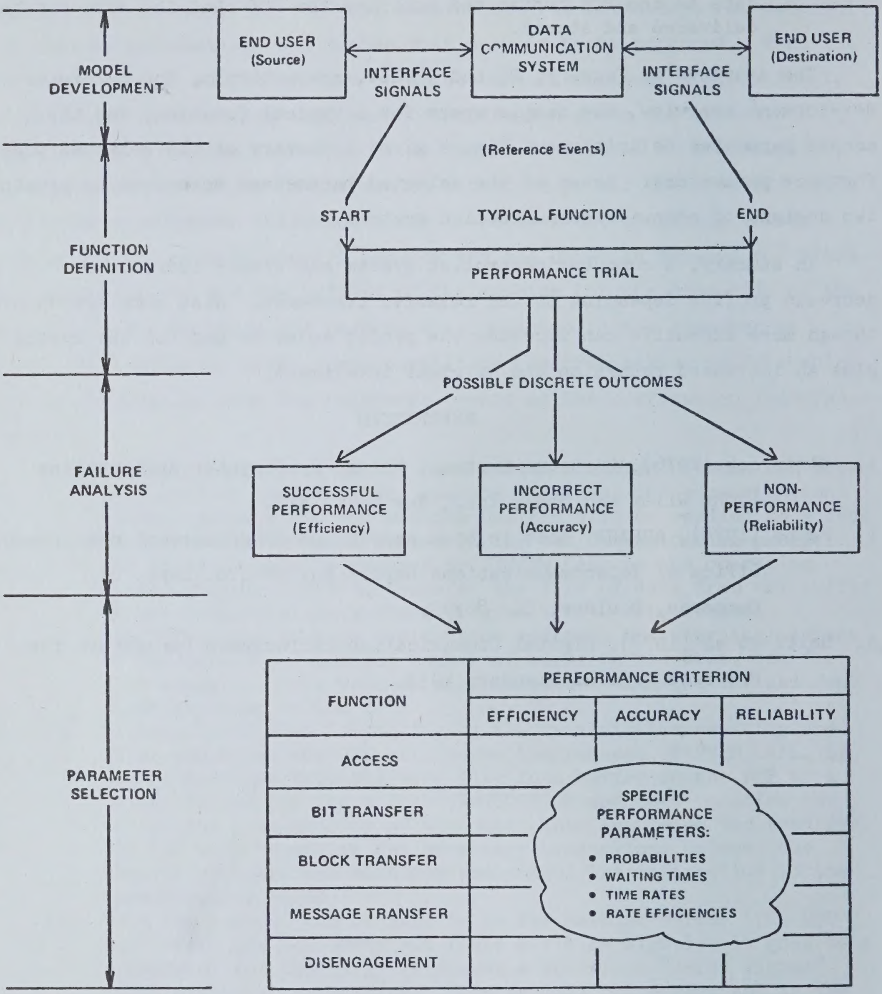

MODEL DEVELOPMENT
FUNCTION DEFINITION
FAILURE ANALYSIS
PARAMETER SELECTION
END USER (Source)
INTERFACE SIGNALS
DATA COMMUNICATION SYSTEM
INTERFACE SIGNALS
END USER (Destination)
(Reference Events)
START
TYPICAL FUNCTION
END
PERFORMANCE TRIAL
POSSIBLE DISCRETE OUTCOMES
SUCCESSFUL PERFORMANCE (Efficiency)
INCORRECT PERFORMANCE (Accuracy)
NON-PERFORMANCE (Reliability)
FUNCTION
PERFORMANCE CRITERION
EFFICIENCY
ACCURACY
RELIABILITY
ACCESS
BIT TRANSFER
BLOCK TRANSFER
MESSAGE TRANSFER
DISENGAGEMENT
SPECIFIC PERFORMANCE PARAMETERS:
• PROBABILITIES
• WAITING TIMES
• TIME RATES
• RATE EFFICIENCIES

Figure 1. Parameter development overview.

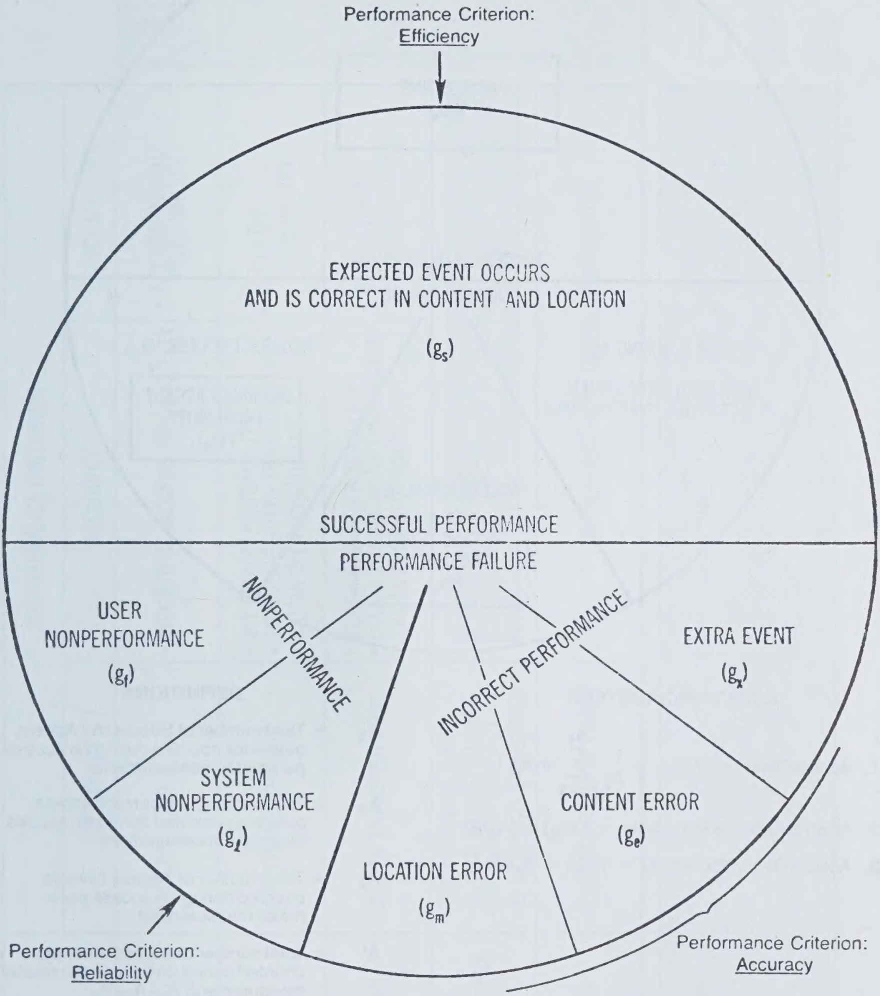
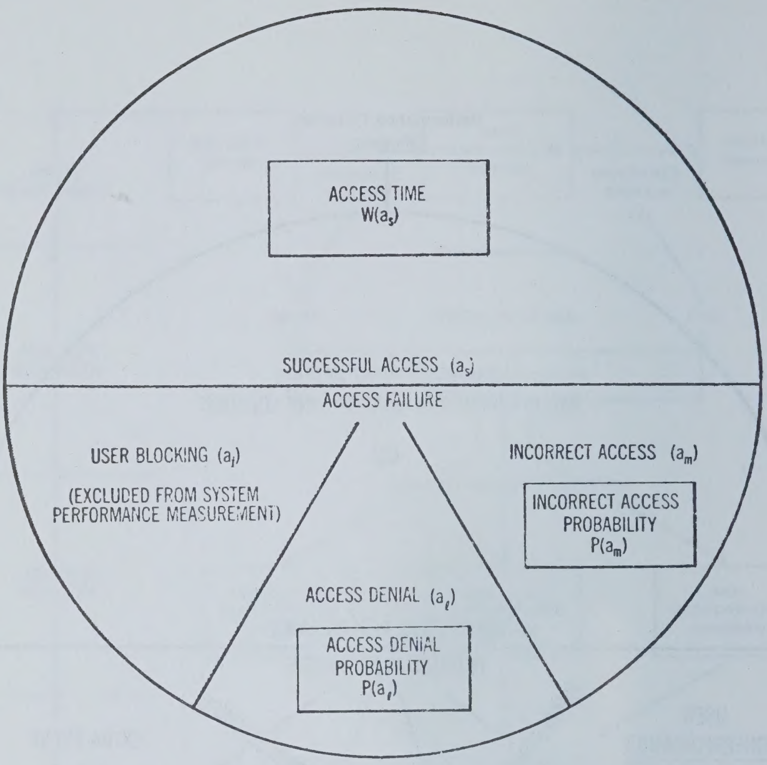

Performance Criterion:
Efficiency
EXPECTED EVENT OCCURS
AND IS CORRECT IN CONTENT AND LOCATION
(g_s)
SUCCESSFUL PERFORMANCE
PERFORMANCE FAILURE
NONPERFORMANCE
INCORRECT PERFORMANCE
USER
NONPERFORMANCE
(g_f)
EXTRA EVENT
(g_x)
SYSTEM
NONPERFORMANCE
(g_l)
CONTENT ERROR
(g_e)
LOCATION ERROR
(g_m)
Performance Criterion:
Reliability
Performance Criterion:
Accuracy

Figure 2. Sample space for a typical primary function.


ACCESS TIME
$W(a_s)$
SUCCESSFUL ACCESS (a_s)
ACCESS FAILURE
USER BLOCKING (a_i)
(EXCLUDED FROM SYSTEM PERFORMANCE MEASUREMENT)
INCORRECT ACCESS (a_m)
INCORRECT ACCESS PROBABILITY
$P(a_m)$
ACCESS DENIAL (a_ℓ)
ACCESS DENIAL PROBABILITY
$P(a_\ell)$

ACCESS PARAMETERS

1. Access Time $= W(a_s) = \frac{1}{A_s} \sum_{a_s=1}^{A_s} w(a_s)$
2. Incorrect Access Probability $= P(a_m) = A_m/A'$
3. Access Denial Probability $= P(a_\ell) = A_\ell/A'$

DEFINITIONS

A_s = Total number of Successful Access outcomes counted during an access parameter measurement.

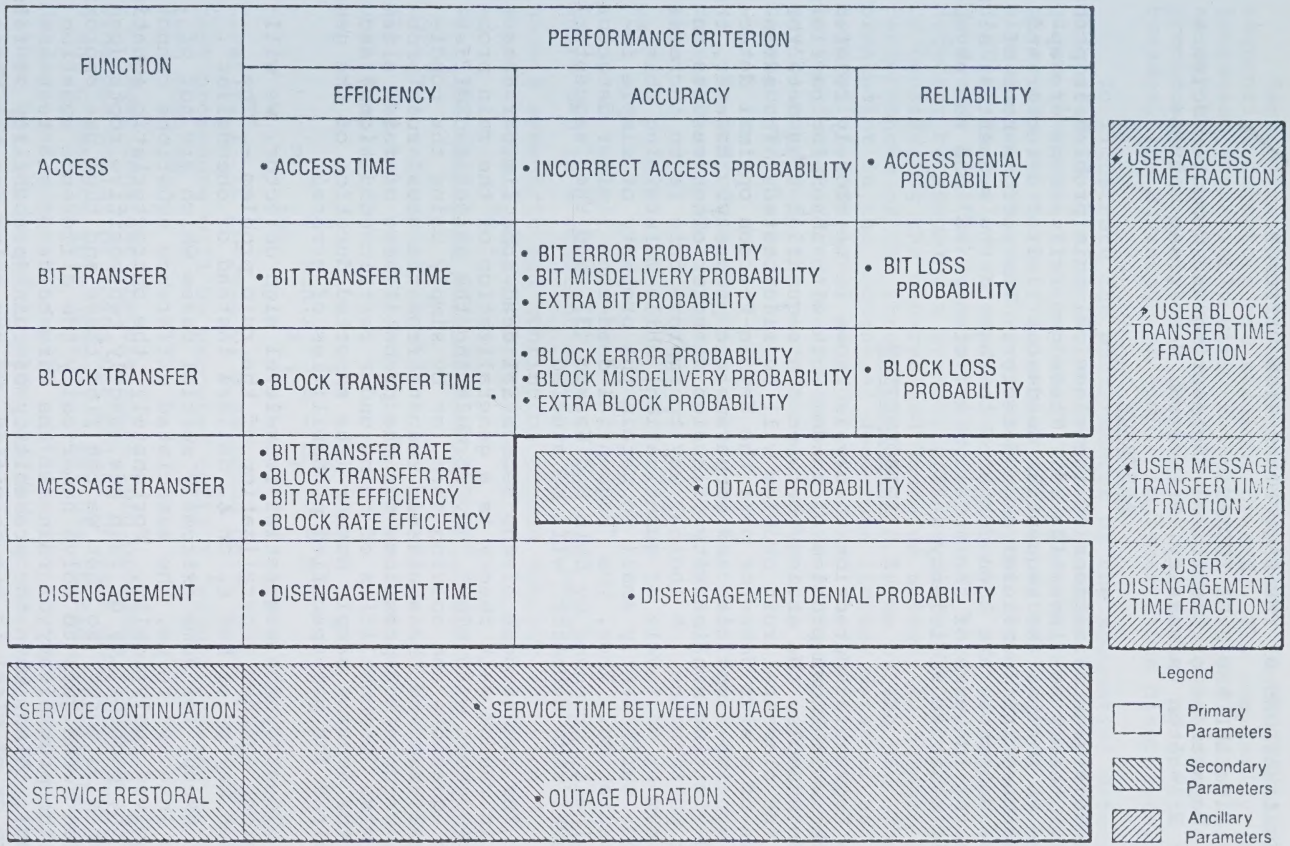
A_m = Total number of Incorrect Access outcomes counted during an access parameter measurement.

A_ℓ = Total number of Access Denials counted during an access parameter measurement.

A' = Total number of access attempts counted during an access parameter measurement: $A_s + A_m + A_\ell$.

$w(a_s)$ = Value of access time measured on a particular successful access attempt.

Figure 3. Access parameter definitions.



FUNCTION	PERFORMANCE CRITERION			
	EFFICIENCY	ACCURACY	RELIABILITY	
ACCESS	• ACCESS TIME	• INCORRECT ACCESS PROBABILITY	• ACCESS DENIAL PROBABILITY	• USER ACCESS TIME FRACTION
BIT TRANSFER	• BIT TRANSFER TIME	• BIT ERROR PROBABILITY • BIT MISDELIVERY PROBABILITY • EXTRA BIT PROBABILITY	• BIT LOSS PROBABILITY	• USER BLOCK TRANSFER TIME FRACTION
BLOCK TRANSFER	• BLOCK TRANSFER TIME	• BLOCK ERROR PROBABILITY • BLOCK MISDELIVERY PROBABILITY • EXTRA BLOCK PROBABILITY	• BLOCK LOSS PROBABILITY	
MESSAGE TRANSFER	• BIT TRANSFER RATE • BLOCK TRANSFER RATE • BIT RATE EFFICIENCY • BLOCK RATE EFFICIENCY	• OUTAGE PROBABILITY		• USER MESSAGE TRANSFER TIME FRACTION
DISENGAGEMENT	• DISENGAGEMENT TIME	• DISENGAGEMENT DENIAL PROBABILITY		• USER DISENGAGEMENT TIME FRACTION
SERVICE CONTINUATION	• SERVICE TIME BETWEEN OUTAGES			
SERVICE RESTORAL	• OUTAGE DURATION			

Legend

- □ Primary Parameters
- ▧ Secondary Parameters
- ▨ Ancillary Parameters

Figure 4. Summary of selected performance parameters.

GENERALIZATIONS OF THE SEQUENTIAL SIGN DETECTOR

C.C. LEE AND J.B. THOMAS
Department of Electrical Engineering and Computer Science
Princeton University, Princeton, New Jersey 08544

ABSTRACT

Some generalizations of the classical ruin problem in probability theory are investigated. These generalizations are applied to analyze the sequential dead-zone limiter detector and the sequential four-level sign detector. The performances of these detectors are then compared to that of the sequential sign detector in terms of the relative efficiency (RE) and the asymptotic relative efficiency (ARE).

I. INTRODUCTION

Sequential detection theory is known to be closely related to the well-known problem of random walk with absorbing barriers. As an example, the analogy between the sequential sign detector and the classical ruin problem will be illustrated. The advantages of a sign detector over the Neyman-Pearson optimal detector are its simplicity and nonparametric property. However, the hard-limiting nonlinearity of a sign detector often results in poor performance. A logical way to improve this is to increase the number of levels of quantization, while maintaining this number sufficiently small to retain the property of simple implementation. Thus, the "sequential dead-zone limiter detector", originally proposed by Shin and Kassam [3], and the "sequential 4-level sign detector" will be studied.

The counterpart of the sequential dead-zone limiter detector in random walk theory is a generalization of the ruin problem - "games with ties". By formulating the associated difference equations and solving them, or by simply using the conditional test and the available results from the usual ruin problem, we obtain expressions for the probabilities of false alarm and miss (probabilities of ruins under certain conditions) and for the average sample number (the expected duration of the game) required for pre-specified probabilities of errors.

To study the sequential four-level sign detector, we will consider another generalization of the ruin problem - "The stakes may be either ℓ_1 or ℓ_2 dollars instead of one dollar". Now, except for some extremely special cases which are not of much interest to us, the associated difference equations cannot be solved analytically. Fortunately, the characteristic equation for the probability of ruin has exactly two positive roots (one of them is unity) so that we can find these and then use approximation techniques to solve numerically the difference equation for the probability of ruin (if the thresholds of the test are known). Also, when the probability of ruin (probability of false alarm or of miss) is pre-specified, we can solve for the thresholds needed. We then state a theorem by which the solution of the difference equation for the expected duration of the game (ASN) can be obtained or approximated with high accuracy without solving the equation.

Generalizations to more-than-four-level input-quantization sequential detectors are straightforward. The complexity is not increased appreciably with the number of levels of quantization, as may be shown by formulating the difference equations and the procedures for analysis. However, as the number of levels increases, the advantage of simple implementation disappears.

To illustrate the performances of the sequential detectors discussed in this paper, we compare the expected sample numbers given the same error probabilities. The ratio of these numbers gives the relative efficiencies (RE), and the limit of RE as the signal strength vanishes gives the asymptotic relative efficiency (ARE). We derive an expression for the ARE of the sequential dead-zone limiter detector with respect to the sequential sign detector by assuming a symmetric noise distribution. Finally, by considering the generalized Gaussian noises, we compare the performances of the three detectors in terms of the RE. The breakpoint in the sequential dead-zone limiter detector and the sequential four-level sign detector, and the quantization levels in the latter, are chosen so as to give optimal performance. As may be expected, the performance improves as the number of quantization levels increases.

To introduce the sign detector, we treat the detection of constant antipodal signals in additive white noise:

$$H: X_i = N_i - s$$

vs.

$$K: X_i = N_i + s$$

where $s>0$ is a constant signal, and the noises N_i are assumed to have a symmetric and continuous common density function f. The observations $X_1, X_2, \ldots$ are assumed to be independent.

II. THE CLASSICAL RUIN PROBLEM AND THE SEQUENTIAL SIGN DETECTOR

The sequential sign detector has been studied in detail in [2]. The test scheme for this detector is the following: At the n-th sample, form

$$T_n \triangleq \sum_{i=1}^{n} \operatorname{sgn}(X_i) \begin{cases} \geq m_2 & \Rightarrow K \\ \leq -m_1 & \Rightarrow H \\ \text{otherwise} & \Rightarrow \text{take one more sample} \end{cases}$$

Clearly, T_n is the sum of a sequence of +1's and -1's, and the test procedure will be executed until T_n either exceeds m_2 or is less than $-m_1$.

Now, consider the classical ruin problem: a gambler either wins one dollar or loses one dollar on each game with probability p and q respectively (p+q=1). Let his initial capital be m_1 dollars and suppose he will continue to play until his capital either has increased to m_1+m_2 dollars or has reduced to zero. In the former case, he wins, and in the later case, he ruins. Let T_n' denote the increment in the gambler's capital; we have the following scheme: After the n-th game,

$$T'_n = \sum_{i=1}^{n} u_i \begin{cases} \geq m_2 & \Rightarrow \text{wins} \\ \leq -m_1 & \Rightarrow \text{ruins} \\ \text{otherwise} & \Rightarrow \text{plays the next game} \end{cases}$$

where u_i is +1 with probability p or -1 with probability q.

This scheme and that of the sequential sign detector are identical if we let

$$p = \Pr\{u_i=+1\} = \Pr\{\text{sgn}(X_i)=+1\} = \Pr\{X_i>0\}$$

and

$$q = \Pr\{u_i=-1\} = \Pr\{\text{sgn}(X_i)=-1\} = \Pr\{X_i<0\}$$

Thus, the probability of false alarm α for the sequential sign detector corresponds to the probability of winning (1-probability of ruin) for the ruin problem with $p=\Pr\{X_i>0|H\}$, and the probability of a miss $(1-\beta)$ corresponds to the probability of ruin with $p=\Pr\{X_i>0|K\}$. In addition, the average sample number (ASN) needed for the sequential sign detector corresponds to the expected duration of the game.

III. THE RUIN PROBLEM WITH "TIES" AND THE SEQUENTIAL DEAD-ZONE LIMITER DETECTOR

In this section, we consider a generalization of the ruin problem - the bet may result in a "tie". The gambler now may either win one dollar or lose one dollar or win (and lose) nothing on each game with probabilities p',q', and r' respectively (p'+q'+r'=1). Let R_{m1} denote the probability that the gambler with initial capital m_1 will ruin. Since, after the first game, the gambler's fortune is either m_1+1 or m_1 or m_1-1 dollars, we have

$$R_{m1} = p'R_{m1+1} + q'R_{m1-1} + r'R_{m1} \qquad (4.1)$$

with boundary conditions $R_0=1$, $R_{m1+m2}=0$. The solution for this difference equation is not difficult [9]. The result is

$$R_{m1} = \left[\left(\frac{q'}{p'}\right)^{m_1+m_2} - \left(\frac{q'}{p'}\right)^{m_1}\right] \Big/ \left[\left(\frac{q'}{p'}\right)^{m_1+m_2} - 1\right] \qquad (4.2)$$

Note that this result is the same as that of the usual ruin problem [1] with p replaced by p' and q by q'.

Next, we consider the duration D_{m1} of the game. To find an expression for D_{m1}, we may solve the associated difference equation:

$$D_{m1} = p'D_{m1+1} + q'D_{m1-1} + r'D_{m1} + 1 \qquad (4.3)$$

with boundary conditions $D_0=0$ and $D_{m1+m2}=0$. The solution is given in [9]. Here we show another approach which is much simpler and will be useful for further applications in the next section. The following theorem is intuitively true, and its proof is given in [9].

Theorem 1: "The expected duration of the game is equal to the expected gain divided by the expected gain in a single trial".

A direct result follows:

Corollary 1: The solution of (4.3) is given by

$$D_{ml} = [R_{ml}(-m_1) + (1-R_{ml})m_2]/(p'-q') \qquad (4.4)$$

This result is the same as that obtained in [9] if we replace R_{ml} by (4.2). Theorem 1 enables us to solve the whole problem by solving the difference equation associated with the probability of ruin.

In the sequential sign detector, we make use of a hard-limiter (Fig. 1) to classify the samples.

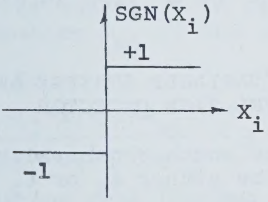


Fig. 1 A hard limiter

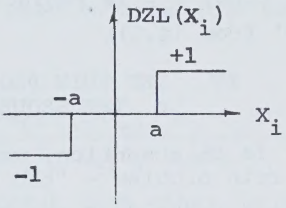


Fig. 2 A symmetric dead-zone limiter with dead-zone (-a,a)

In some cases, poor performance may result. If we use a dead-zone limiter (Fig. 2) instead, that is, if we eliminate a class of samples (those that lie in the dead-zone), the resulting detector may have significantly better performance characteristics. This sequential dead-zone limiter detector was originally proposed and analyzed, based on conditional tests, by Shin and Kassam [3]. We now use the ruin problem with ties to investigate this detector. The test scheme for a sequential dead-zone limiter detector is the following: At the n-th step

$$T_n = \sum_{i=1}^{n} DZL(X_i) \begin{cases} \geq \; m_2 & \Rightarrow K \\ \leq - m_1 & \Rightarrow H \\ \text{otherwise} & \Rightarrow \text{take one more sample} \end{cases}$$

where $DZL(X_i)$ has three possible values, +1,0, and -1. If we let

$$\begin{aligned} p' &= \Pr\{DZL(X_i)=+1 | H\} = \Pr\{X_i>a | H\} = 1-F(a+s) \\ &= F(-a-s) = \Pr\{X_i<-a | K\} = \Pr\{DZL(X_i)=-1 | K\} \end{aligned} \qquad (4.5)$$

and

$$\begin{aligned} q' &= \Pr\{DZL(X_i)=-1 | H\} = \Pr\{X_i<-a | H\} = F(s-a) \\ &= 1-F(a-s) = \Pr\{X_i>a | K\} = \Pr\{DZL(X_i)=+1 | K\} \end{aligned} \qquad (4.6)$$

where F is the noise distribution function, the analogy between the ruin problem with ties and the sequential dead-zone limiter detector becomes apparent. It thus follows that

$$\begin{aligned} \alpha &= \text{Probability of win with parameters } p' \text{ and } q' \\ &= 1 - [(\tfrac{q'}{p'})^{m_1+m_2} - (\tfrac{q'}{p'})^{m_1}]/[(\tfrac{q'}{p'})^{m_1+m_2}-1] \end{aligned} \qquad (4.7)$$

and

$$1-\beta = \text{Probability of ruin with parameters } q' \text{ and } p'$$

$$= \left[\left(\frac{p'}{q'}\right)^{m_1+m_2} - \left(\frac{p'}{q'}\right)^{m_1}\right] \Big/ \left[\left(\frac{p'}{q'}\right)^{m_1+m_2} - 1\right] \quad (4.8)$$

If α and $1-\beta$ are pre-specified, we may solve for m_1 and m_2:

$$m_1 = \ell og[(1-\beta)/(1-\alpha)]/\ell og(q'/p') \quad (4.9)$$

and

$$m_2 = \ell og(\alpha/\beta)/\ell og(q'/p') \quad (4.10)$$

Substituting these expressions into (4.4), we obtain the average sample number E(N). If H is true, R_{m1} is replaced by $1-\alpha$. If K is true, R_{m1} is replaced by $1-\beta$, p' by q' from (4.6), and q' by p' from (4.5).

IV. THE RUIN PROBLEM WITH "VARIABLE STAKES" AND n THE SEQUENTIAL 4-LEVEL SIGN DETECTOR

In this section, we investigate another generalization of the ruin problem - "the stakes may be either ℓ_1 or ℓ_2 ($\ell_2 > \ell_1$) dollars instead of one dollar". The gambler may either win ℓ_1 dollars, or win ℓ_2 dollars, or lose ℓ_1 dollars, or lose ℓ_2 dollars on each game with probabilities p_1, p_2, p_{-1}, p_{-2} respectively ($p_1+p_2+p_{-1}+p_{-2}=1$). Then, with initial capital m_1, the probability of ruin R_{m1} satisfies

$$R_{m1} = p_1 R_{m1+\ell 1} + p_2 R_{m1+\ell 2} + p_{-1} R_{m1-\ell 1} + p_{-2} R_{m1-\ell_2} \quad (5.1)$$

with boundary conditions

$$\begin{aligned} R_n &= 1 \quad , -\ell_2 + 1 \le n \le 0 \\ R_n &= 0 \quad , m_1 + m_2 \le n \le m_1 + m_2 + \ell_2 - 1 \end{aligned} \quad (5.2)$$

Unfortunately, this difference equation cannot be solved analytically except for some extremely special cases (e.g. $p_1=p_2=p_{-1} =p_{-2}=1/4$) which are not of great interest to us. But the situation is not too bad. If we substitute $R_m=x^m$ into the equation, we obtain its characteristic equation:

$$p_2 x^{2\ell_2} + p_1 x^{\ell_1+\ell_2} - x^{\ell_2} + p_{-1} x^{\ell_2-\ell_1} + p_{-2} = 0 \quad (5.3)$$

Clearly, unity is one of the roots of (5.3). Since the coefficients on the left side of (5.3) change signs twice, there is exactly one more positive root x_1. The x_1 can be located numerically [9] and then the solution of (5.1) can be bounded [9] as follows:

$$\frac{x_1^{m_1+m_2+\ell_2-1} - x_1^{m_1+\ell_2-1}}{x_1^{m_1+m_2+\ell_2-1} - 1} \le R_{m1} \le \frac{x_1^{m_1+m_2+\ell_2-1} - x_1^{m_1}}{x_1^{m_1+m_2+\ell_2-1} - 1} \quad (5.4)$$

Equation (5.4) can be further simplified to obtain, approximately

$$m_1 \le \ell og R_{m1}/\ell og x_1 \le m_1+\ell_2-1 \qquad (5.5)$$

when m_1 and m_2 are large compared to ℓ_2 and the expected gain in a single trial is positive. The latter condition is what we are interested in when we study later the Sequential 4-Level Sign Detector. Since $m_1 \gg \ell_2$, we may take $\ell og R_{m1}/\ell og x_1 \doteq (2m_1+\ell_2-1)/2$ or

$$R_{m1} \doteq \exp[(2m_1+\ell_2-1)\ell og x_1/2] \qquad (5.6)$$

Simulations show that this approximation is very accurate (to the fifth decimal when $m_1 > 100\ \ell_2$).

Once the probability of ruin R_{m1} is obtained, the following corollary of Theorem 1 in Section 4 is used to bound the expected duration D_{m1} of the game.

Corollary 2:

$$\frac{R_{m1}(-m_1-\ell_2+1)+(1-R_{m1})m_2}{p_1\ell_1+p_2\ell_2-p_{-1}\ell_1-p_{-2}\ell_2} \le D_{m1} \le \frac{R_{m1}(-m_1)+(1-R_{m1})(m_2+\ell_2-1)}{p_1\ell_1+p_2\ell_2-p_{-1}\ell_1-p_{-2}\ell_2}$$

It is apparent that the two bounds approach each other as m_1 and m_2 become very large compared to ℓ_2 as is usually the case in our applications.

As indicated in Sec. 1, to improve the poor performance caused in some cases by the hard-limiting nonlinearity of a sign detector, we may increase the levels of input-quantization. In [4], Kassam and Thomas proposed a 4-level sign detector which uses the nonlinearity shown in Fig. 3.

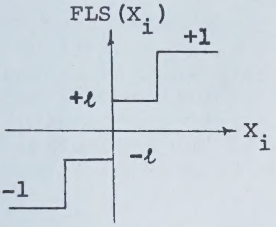


Fig. 3 Normalized 4-level quantizer

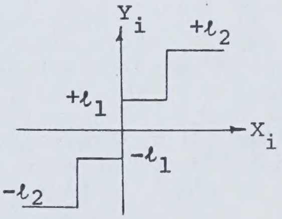


Fig. 4 4-level quantizer

We now investigate the sequential counterpart of this detector. We impose the following condition: $\ell = \ell_1/\ell_2$ is a rational number $(\ell_2 > \ell_1)$ so that the nonlinearity in Fig. 3 is equivalent to that in Fig. 4 if we properly adjust the test thresholds. Thus, the test scheme for a sequential 4-level sign detector is: at the n-th step,

$$T_n = \sum_{i=1}^{n} FLS(X_i) \begin{cases} \ge \ m_2 & \Rightarrow K \\ \le -m_1 & \Rightarrow H \\ \text{otherwise} & \Rightarrow \text{take one more sample.} \end{cases}$$

where $FLS(X_i)$ has four possible values: $-\ell_2$, $-\ell_1$, $+\ell_1$ and $+\ell_2$. If we let

$$p_2=\Pr\{FLS(X_i)=+\ell_2|K\}=\Pr\{X_i>a|K\}=1-F(a-s)$$

$$=F(s-a)=\Pr\{X_i<-a|H\}=\Pr\{FLS(X_i)=-\ell_2|H\}$$

$$p_1=\Pr\{FLS(X_i)=+\ell_1|K\}=F(a-s)-F(-s)=F(s)-F(s-a)$$

$$=\Pr\{FLS(X_i)=-\ell_1|H\}$$

$$p_{-1}=\Pr\{FLS(X_i)=-\ell_1|K\}=F(-s)-F(-a-s)=F(s+a)-F(s)$$

$$=\Pr\{FLS(X_i)=+\ell_1|H\}$$

$$p_{-2}=\Pr\{FLS(X_i)=-\ell_2|K\}=F(-a-s)=1-F(a+s)=\Pr\{FLS(X_i)=+\ell_2|H\}$$

the analogy between this detector and the generalization of the ruin problem becomes obvious. We thus have

$$\alpha = \Pr\{\text{decide } K|H\} = \Pr\{\text{ruin}|\text{capital } m_2\} = R_{m2}$$

$$1-\beta = \Pr\{\text{decide } H|K\} = \Pr\{\text{win}|\text{capital } m_1\} = R_{m1}$$

For prespecified α and β, we may obtain m_1 and m_2 by using (5.5). Then, by applying Corollary 2, the average sample numbers (ASN) are obtained: $E(N|H) = D_{m2}$ and $E(N|K) = D_{m1}$.

V. PERFORMANCES

A. The Relative Efficiency (RE) and the Asymptotic Relative Efficiency (ARE) of the Sequential Dead-Zone Limiter Detector With Respect to the Sequential Sign Detector

The detection problem considered in this paper, as stated in Sec. 2, is "constant antipodal signals in additive white noise". Let f and F denote the density and the distribution function of the noise. By imposing only the condition that f is continuous and symmetric (with respect to the origin), we can derive [9] an expression for the RE of the sequential dead-zone limiter detector with respect to the sequential sign detector. The result is: under H and K.

$$\mathrm{RE21} = \frac{\mathrm{ASN}_1}{\mathrm{ASN}_2} = \frac{[1-2F(s+a)]\log\{F(s+a)/[1-F(s+a)]\}}{[1-2F(s)]\log\{F(s)/[1-F(s)]\}} \tag{6.1}$$

Certainly, we have to pick an optimal value for "a" to maximize $\mathrm{RE}_{2,1}$. The generalized Gaussian density function [5]

$$f(x) = \frac{c\sigma^{-1}v(c)}{2\Gamma(1/c)}\exp[-(\sigma^{-1}v(c)|x|)^c] \tag{6.2}$$

represents a class of distributions with various rates of exponential decay, so it may be instructive to choose it as the noise distribution to check the performances of the detectors discussed in this paper. Tables 1 through 5 list RE21 for various values of c with α as a variable. The greater is c, the greater the RE21 that can be expected.

The ARE, defined as the limit of the RE as the signal strength goes to zero, is

$$\mathrm{ARE21} = f^2(a)/[2f^2(0)F(-a)] \quad (6.3)$$

The derivation of (6.3) is also shown in [9]. Note that this equation is identical to that found in [3] and [4]. Fig. 5 shows this ARE for generalized Gaussian noise with "a" as a variable. We can see from the figure that, in the Gaussian case (c=2), the ARE may achieve a value greater than 1.27. Thus, the improvements made by the dead-zone limiter detector is significant.

B. Comparisons of the Sequential Dead-Zone Limiter Detector and The Sequential 4-Level Sign Detector with the Sequential Sign Detector in Terms of the RE

Analytic expressions for the RE's of the sequential 4-level sign detector (w.r.t. the sequential sign detector) are not available. Numerical results for this detector are obtained for the following three cases:

(i) $\ell_1=1$, $\ell_2=2$ (Detector #3)

(ii) $\ell_1=1$, $\ell_2=3$ (Detector #4)

(iii) $\ell_1=1$, $\ell_2=4$ (Detector #5)

Tables 1 through 5 give the RE's of Detector #2 through Detector #5 with respect to Detector #1 (the sign detector). Table 3 is for the Gaussian noise in which the RE for the corresponding sequential linear detector (Detector #6), which is Neyman-Pearson optimal for Gaussian noise, is also included for convenience of comparisons.

These tables show significant improvements over the sequential sign detector. The sequential 4-level sign detector performs better than the dead-zone limiter detector, as may be expected. As a matter of fact, the sequential dead-zone limiter detector is, of course, a degenerate sequential 4-level sign detector ($\ell_1=0$, $\ell_2=1$).

ACKNOWLEDGEMENTS

The authors acknowledge the support of the Air Force Office of Scientific Research under Grant AFOSR-76-3083B.

REFERENCES

1. W. Feller, An Introduction to Probability Theory and Its Applications, Vol. I, John Wiley & Sons, New York, 1968.
2. S. Tantaratana, On the Relative Efficiencies of Some Parametric and Nonparametric Sequential Detectors, Ph.D. Dissertation, Dept. of EECS, Princeton University, 1977.
3. J.G. Shin and S.A. Kassam, "Sequential Detection Based on Quantization and Conditional Testing", Proceedings of the 1979 Conference on Information Sciences and Systems, The Johns Hopkins University, Baltimore, Md., March 1978; pp.487-491.
4. S.A. Kassam and J.B. Thomas, "A Modified Sign Detector Based on a Four-Level Input Quantization", presented at the 13th Annual Allerton Conference on Circuit and System Theory, Oct. 1-3, 1975.
5. J.H. Miller and J.B. Thomas, "Detectors for Discrete-Time Signals in Non-Gaussian Noise", IEEE Trans. Inform. Theory, Vol. IT-18, No. 2, March 1972.

6. S.A. Kassam and J.B. Thomas, "The Dead-Zone Limiter - An Application of Conditional Tests in Nonparametric Detection", J. Acous. Soc. of Amer., Vol. 60, pp.857-862, Oct. 1976.
7. J.B. Thomas, "Nonparametric Detection", Proc. of the IEEE, Vol. 58, No. 5, May 1970.
8. S.A. Kassam and J.B. Thomas, "Generalizations of the Sign Detector Based on Conditional Tests", IEEE Trans. Comm., Vol. COM-24, No. 5, May 1976.
9. C.C. Lee, Ph.D. Dissertation, Dept. of EECS, Princeton University, in preparation.

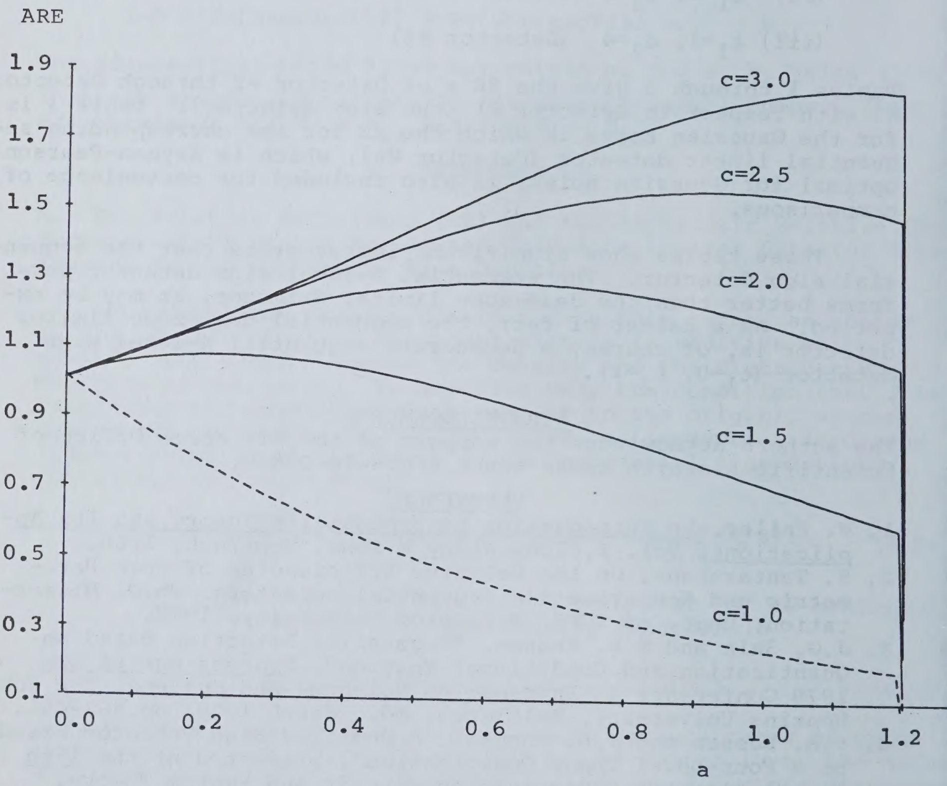


Fig. 5 The ARE of the sequential dead-zone limiter with respect to the sequential sign detector. Generalized Gaussian noises.

s/σ	0.1		0.01	
α	10^{-2}	10^{-6}	10^{-2}	10^{-6}
RE21	1.031	1.031	1.003	1.004
RE31	1.051	1.040	1.005	1.004
RE41	1.058	1.044	1.006	1.005
RE51	1.044	1.039	1.005	1.004

Table 1

c=1.0 (Laplace Noise)

s/σ	0.1		0.01	
α	10^{-2}	10^{-6}	10^{-2}	10^{-6}
RE21	1.080	1.080	1.065	1.065
RE31	1.162	1.151	1.141	1.139
RE41	1.159	1.145	1.130	1.128
RE51	1.131	1.126	1.113	1.113

Table 2

c=1.5

s/σ	0.1		0.01	
α	10^{-2}	10^{-6}	10^{-2}	10^{-6}
RE21	1.273	1.273	1.270	1.270
RE31	1.342	1.329	1.324	1.323
RE41	1.414	1.395	1.387	1.385
RE51	1.392	1.385	1.383	1.382
RE61	1.573	1.573	1.571	1.571

Table 3

c=2 (Gaussian Noise)

s/σ	0.1		0.01	
α	10^{-2}	10^{-6}	10^{-2}	10^{-6}
RE21	1.531	1.531	1.533	1.533
RE31	1.492	1.479	1.474	1.473
RE41	1.660	1.635	1.628	1.625
RE51	1.675	1.667	1.665	1.664

Table 4

c=2.5

s/σ	0.1		0.01	
α	10^{-2}	10^{-6}	10^{-2}	10^{-6}
RE21	1.815	1.815	1.818	1.818
RE31	1.619	1.604	1.600	1.598
RE41	1.881	1.852	1.843	1.840
RE51	1.944	1.934	1.931	1.930

Table 5

c=3.0

Suboptimum Detection Schemes with Counting Point Process Observations

A. A. Lazar and S. C. Schwartz

Department of Electrical Engineering and Computer Science
Princeton University
Princeton NJ 08544

ABSTRACT

Several suboptimum detection procedures with counting point process observations are discussed. Attention is focused on the analysis and implementation of a new functional form for the random intensity rate of the observation process.

I. Introduction

The problem of modelling and subsequently detecting the presence or absence of an incident field in background radiation using point process observations has been extensively studied in various degrees of generality [1], [3], [5], [11], [16]. Experimental results [2] for estimating the random intensity rate (RIR) of the observation process have shown a good deal of agreement between the more sophisticated theoretical results (MMSE estimates) and their possible practical implementation.

The majority of the existing optimal detection schemes are based on the fact that the rate of the arrivals of photoelectrons at the input of the detector is proportional to the intensity of the incident field, *i.e.*,

$$\lambda_t = \eta |X_t|^2, \tag{1}$$

where X_t is the amplitude of the incident field, η is a constant related to the photodetection device and λ is the rate of arrivals of photoelectrons [3], [5], [11], [16]. It is clear that the transformation given in (1) affects the "information" available at the input of the detector.

Snyder-Rhodes [15] have given a Cramér-Rao type lower bound for estimating a Gaussian process with doubly stochastic point process (DSPP) observations. The RIR was defined as

$$\lambda_t = \lambda_t(X_t). \tag{2}$$

It is not apparent, however, what specific functional form the intensity rate should take so as to achieve this lower bound [16, p. 373].

In our study, the classical hypothesis testing problem

$$\begin{aligned} &H_0\text{: under } P_0,\ X_t \quad \text{is a Wiener process} \\ &H_1\text{: under } P_1,\ X_t - \int_0^t S_\tau d\tau \quad \text{is a Wiener process,} \end{aligned} \tag{3}$$

where $X = \{X_t, A_t\}$, $0 \leqslant t \leqslant T$, are the continuous-time observations is transformed into a hypothesis testing problem

$$\begin{aligned} &H_0\text{: under } \tilde{P}_0,\ N_t \text{ is a DSPP with RIR } \lambda_t^{(0)} \\ &H_1\text{: under } \tilde{P}_1,\ N_t \text{ is a DSPP with RIR } \lambda_t^{(1)}, \end{aligned} \tag{4}$$

where $N=\{N_t, A_t \otimes \tilde{A}_t\}$, $0 \leqslant t \leqslant T$, are the counting point process observations. One can think of this formulation as one in which one of the two possible continuous-time signals modulates an optical communication system, with the resulting observations a counting process.

One of the aims of our investigation is to represent the RIR λ as a functional of the "information bearing process" and the observations, *i.e.*,

$$\lambda_t = \lambda_t(X_\tau, N_\tau, \tau \leqslant t) \tag{5}$$

such that we attain (or come close to) "optimal properties" (which have yet to be specified) for the likelihood ratio of the hypothesis testing problem (4).

Previous investigations [7], [8] have shown that a natural choice for the RIR is

$$\lambda_\sigma = E_0\left\{ \frac{dP_1}{dP_0} \mid A_\sigma \right\}, \tag{6}$$

i.e., the Radon-Nikodym derivative of the corresponding measures P_1 and P_0. This is just the likelihood ratio of the hypothesis testing problem (3). Following [7], [8] the choice given in (6) for λ_σ will be called in the sequel the most natural choice (MNC) of the RIR.

Suboptimum procedures are investigated since it does not make sense to implement (6). That is, if we had (6), the detection problem could be solved in the original form given by (3), rather than trying to find a transformation to take the hypothesis testing problem into the form given by (4).

This paper is organized as follows. In section II we present some preliminaries used throughout this paper. Two suboptimum schemes for the detection of a Gaussian signal in white Gaussian noise having counting point process observations are discussed in section III. It is shown that the previous choice (1) for the RIR can be viewed as an approximation for the log-likelihood ratio (given in (7)). In section IV we present exact time statistics for the detection schemes for the problem of detecting a "Gaussian optical field" in background radiation having DSPP observations.

II. Preliminaries

Let us assume that the stochastic process $S=\{S_t, A_t\}$, $0 \leqslant t \leqslant T$, defined on (Ω, A, P_1) is Gaussian with zero mean and autocorrelation function

$$R_{SS}(t,s) = \sum_{i \in \mathrm{N}} \gamma_i \phi_i(t) \phi_i(s),$$

where the $\{\gamma_i\}_{i \in \mathrm{N}}$ are the eigenvalues of the associated integral equation [11]. For the likelihood ratio

$$\Lambda_\sigma = E_0\left\{ \frac{dP_1}{dP_0} \mid A_\sigma \right\}$$

for all σ, $0 \leqslant \sigma \leqslant T$, several implementations are known; perhaps the best known being the estimator-correlator.

We will use in the following, however, the equivalent filter-squarer-integrator receiver given by

$$\ln \Lambda_\sigma = D_F^{-1}(1) \int_0^\sigma Y_\tau^2 d\tau, \tag{7}$$

where

$$D_F(1) = \prod_{i \in N} (1 + \gamma_i)$$

is the Fredholm determinant and

$$Y_\tau = \int_0^\tau g(\tau,u) X_u du$$

is a filtered version of the observations, where $g(\tau,u)$ is a causal filter (see [17] fore more details). The block diagram of the filter-square-integrator receiver is shown in Fig.1.

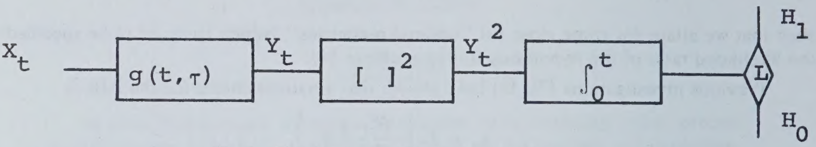


Fig.1 Filter-square-integrator receiver.

Note that the causal filter $g(t,\tau)$ reduces the hypothesis testing problem (3) to

$$\begin{aligned} &H_0\text{: under } P_0,\ Y_t \text{ is Gaussian with autocorrelation function } h(t,s) \\ &H_1\text{: under } P_1,\ Y_t \text{ is Gaussian with autocorrelation function } R_{SS}(t,s), \end{aligned} \tag{8}$$

where $h(t,s)$ and $R_{SS}(t,s)$ are related through the Fredholm equation of the second kind

$$h(t,s) + \int_0^T h(t,u) R_{SS}(u,s) du = R_{SS}(t,s)$$

for all t, s, $0 \leqslant t,s \leqslant T$. In the following we will use the property that the log of the likelihood ratio in (7) is positive.

III. Suboptimum Schemes

It is apparent that it does not make any sense to implement the processor specified by (see also (6))

$$\lambda_\sigma = E_0 \left\{ \frac{dP_1}{dP_0} \mid A_\sigma \right\} \tag{9}$$

for all σ, $0 \leqslant \sigma \leqslant T$, since once this is done, the detection problem as given in (3) is immediately solved. λ_T is the required sufficient statistic and there is no justification for considering a transformation to a point process. In this section we will examine possible "suboptimum" solutions to the hypothesis testing problem (3) by finding an approximate real time implementation for the MNC of the RIR λ given by (9). Motivated by (7), let us define the RIR

$$\lambda_\sigma = \eta \mid Y_\sigma \mid^2 = \eta \left| \int_0^\sigma g(\sigma,\tau) X_\tau d\tau \right|^2, \tag{10}$$

where η is a constant which allows a certain degree of freedom in the actual implementation. Note that this choice of the RIR assures us that equivalent Gaussian hypothesis testing problems with (8) (the existence of such problems can easily be inferred from [6]) are transformed into equivalent point process detection problems. This point is made clear by (10).

The "suboptimum processor - optimum detector" detection procedure is illustrated in the block diagram in Fig.2. Recall that, [7, Theorem 1], the detector estimates (under the

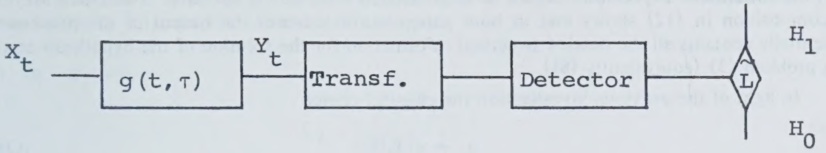


Fig.2 Suboptimum processor - optimum detector.

hypothesis H_0) the input likelihood ratio.

The Doob-Meyer decomposition [9] of the point process $N=\{N_t, A_t\otimes\tilde{A}_t\}$, $0\leqslant t\leqslant T$, (for a precise mathematical setting see [7],[8]) which is given by:

$$N_t = \int_0^t \lambda_\sigma d\sigma + M_t = C_t + M_t \tag{11}$$

for all t, $0\leqslant t\leqslant T$, suggests a different suboptimum solution to the detection problem (8). If we set

$$\lambda_\sigma = \eta\,|Y_\sigma|^2$$

the above decomposition becomes:

$$N_t = \eta\int_0^t |Y_\sigma|^2 d\sigma + M_t = \xi \ln\Lambda_t + M_t, \tag{12}$$

where $\xi=\eta D_F(1)$. The block diagram for the Doob-Meyer decomposition (12) is shown in Fig.3.

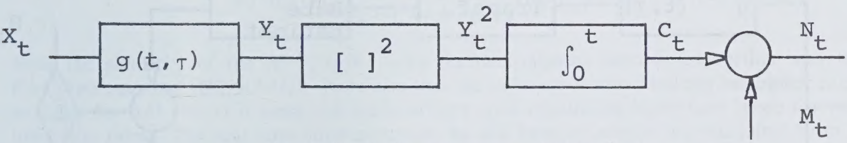


Fig.3 Block diagram representation of the Doob-Meyer decomposition.

Comparing the relations (7) to (12) or correspondingly the block diagrams in Fig.1 and Fig.3, we note that in this case in the observation counting point process $N=\{N_t, A_t\otimes\tilde{A}_t\}$, $0\leqslant t\leqslant T$, the input log-likelihood ratio process $\ln\Lambda=\{\ln\Lambda_t, A_t\otimes\tilde{A}_0\}$, $0\leqslant t\leqslant T$, is embedded in the "additive noise" process $M=\{M_t, A_t\otimes\tilde{A}_t\}$, $0\leqslant t\leqslant T$ (see [7], [8] for notations).

Thus the "suboptimum processor - suboptimum detector" detection procedure will be given by the block diagram in Fig.4.

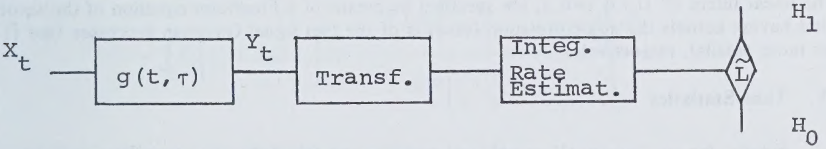


Fig.4 Suboptimum processor - suboptimum detector.

Note that the difference between the two suboptimum schemes (Fig.2 and Fig.4) is given by the fact that in general:

$$\bar{P}_0\left\{\bar{E}_0\left\{\ln \frac{dP_1}{dP_0} \mid A_0\otimes\tilde{A}_T\right\} - \ln \bar{E}_0\left\{\frac{dP_1}{dP_0} \mid A_0\otimes\tilde{A}_T\right\} \neq 0\right\} > 0,$$

i.e., the conditional expectation cannot be interchanged with the log operator. The Doob-Meyer decomposition in (12) shows that in both suboptimum schemes the output of the processor essentially contains all the needed structural information for the solution of the hypothesis testing problem (3) (equivalently (8)).

In light of the previous investigation the classical choice

$$\lambda_t = \eta |X_t|^2 \tag{2}$$

for the RIR, can be viewed as an approximation to (10). Note that with this RIR the first term of the Doob-Meyer decomposition (in (12)) will merely be an approximation of the log-likelihood ratio of the input.

Remark. The likelihood ratio of the hypothesis testing problem

$$\begin{aligned} H_0&:\ \text{under } P_0,\ X_t - \int_0^t S_\tau^{(0)} d\tau \quad \text{is a Wiener process} \\ H_1&:\ \text{under } P_1,\ X_t - \int_0^t S_\tau^{(1)} d\tau \quad \text{is a Wiener process} \end{aligned} \tag{13}$$

(where $S^{(i)}=\{S_t^{(i)},A_t\}$, $0 \leqslant t \leqslant T$, $i=0,1$, are (signal) Gaussian processes of given autocorrelation function under the probability measures P_i, $i=0,1$, respectively) does not admit, in general, a filter-squarer-integrator receiver implementation (see [17]). In such a situation, following along lines similar to that above, the corresponding suboptimum schemes can be also easily suplied. In Fig.5 the "suboptimum processor - suboptimum detector" decision scheme is given.

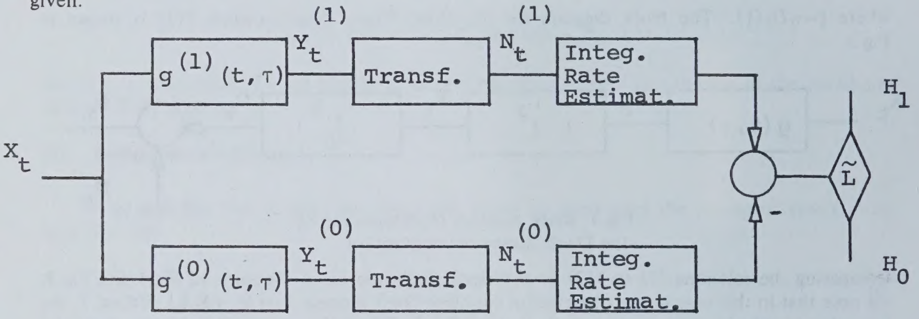


Fig.5 Suboptimum processor - suboptimum detector for the two signal detection problem.

The linear filters $g^{(i)}(t,\tau)$, $i=0,1$, are specified by means of a Fredholm equation of the second kind having kernels the autocorrelation function of the two signal Gaussian processes (see [17] for more details), respectively.

IV. Time Statistics

Finally, for an "intuitive" comparison between the two different suboptimum procedures presented above, we now present exact time statistics of the detection schemes for the problem of detecting an optical field (modelled as a Gaussian process) in background radiation having DSPP observations. First we will consider the "suboptimum processor - optimum detector" detection scheme.

The sufficient statistics of the detector can be specified as follows (the computations are done following [10], see also [4] and [16]):

(i) if $N_T = 0$

$$\tilde{\Lambda}_T = \prod_{i \in \mathrm{N}} \frac{1 + \eta\, \gamma_i}{1 + \eta\, \dfrac{\gamma_i}{1 + \gamma_i}}. \tag{14}$$

(ii) if $N_T \geqslant 1$

$$\tilde{\Lambda}_T = \prod_{i \in \mathrm{N}} \frac{1 + \eta\, \gamma_i}{1 + \eta\, \dfrac{\gamma_i}{1 + \gamma_i}} . \frac{H^{(1)}\begin{bmatrix} t_1 & t_2 & \cdots & t_n \\ t_1 & t_2 & \cdots & t_n \end{bmatrix}}{H^{(0)}\begin{bmatrix} t_1 & t_2 & \cdots & t_n \\ t_1 & t_2 & \cdots & t_n \end{bmatrix}}, \tag{15}$$

where the set of the $\{t_i\}$, $1 \leqslant i \leqslant n$, represents the arrivals of the point process $N = \{N_t, \tilde{A}_t\}$, $0 \leqslant t \leqslant T$. The $H^{(i)}$, $i = 0, 1$ are given by

$$H^{(i)}\begin{bmatrix} t_1 & t_2 & \cdots & t_n \\ t_1 & t_2 & \cdots & t_n \end{bmatrix} = \sum_{\eta \in \mathbf{P}_n} h^{(i)}(t_1, t_{\eta_1})\, h^{(i)}(t_2, t_{\eta_2}) \quad \cdots \quad h^{(i)}(t_n, t_{\eta_n}),$$

where $\mathbf{P}_n$ is the set of all n! permutations of the integers $\{1,2,\ldots,n\}$ and

$$h^{(i)}(t,s) = \sum_{k \in \mathrm{N}} \frac{\gamma_k}{1 + (\eta + \delta_{0i})\gamma_k}\, \phi_k(t)\, \phi_k(s).$$

Note the similarity of the $H^{(i)}$, $i = 0, 1$ with the determinants used in connection with the Fredholm equation ([12, p.172]). Note also that the sufficient statistic will not be implemented as given by (14) and (15) since this leads to very time consuming algorithms (even for very low count rates). For real time implementation we will have to consider an equivalent form of the likelihood ratio of the hypothesis testing (4) and proceed analogously to [13].

Next let us consider the time statistics of the "suboptimum processor - suboptimum detector" detection scheme for the same problem. The computations are based on [4].

$$\hat{C}_T = \bar{E}_0\left\{ C_T \mid A_0 \otimes \tilde{A}_T \right\} = \bar{E}_0\left\{ \int_0^T \lambda_\sigma\, d\sigma \mid A_0 \otimes \tilde{A}_T \right\} =$$

$$= \int_0^T \bar{E}_0\left\{ \lambda_\sigma \mid A_0 \otimes \tilde{A}_T \right\} d\sigma = \frac{\displaystyle\int_0^T H^{(0)}\begin{bmatrix} \tau & t_1 & \cdots & t_n \\ \tau & t_1 & \cdots & t_n \end{bmatrix} d\tau}{H^{(0)}\begin{bmatrix} t_1 & t_2 & \cdots & t_n \\ t_1 & t_2 & \cdots & t_n \end{bmatrix}}.$$

The real time implementation is similar to that given in [14].

V. Conclusion

In this paper we have investigated the possibilities of using a more general functional form for the RIR in detection problems with counting point process observations.

The MNC for the RIR suggests possible suboptimum implementations for the processor and subsequently for the detector. A new functional form for the RIR was proposed and its implementation specified. We have also indicated that the RIR previously used in practical implementations can be viewed as an approximation to the input log-likelihood ratio. The simulation of the two suboptimum schemes presented here is being carried out, and the results will be reported elsewhere.

Acknowledgement

The research reported here was supported in part by the National Science Foundation under grant ENG-75-09610.

References

[1] Bar-David, I., "Communication Under Poisson Regime", *IEEE Trans. Inf. Theory,* Vol. **IT-15,** No. 1, January 1969.

[2] Davidson, F. - Romanski, J., "Experimental Performance of Point-Process Estimation Procedures for Log-Normal Fading", *IEEE Trans. Inf. Theory,* Vol. **IT-22,** No. 3, May 1972.

[3] Gagliardi, R.M. and Karp, S., *Optical Communications,* John Wiley, New York, 1976.

[4] Grandell, J., *Doubly Stochastic Poisson Processes,* Lecture Notes in Mathematics, Vol. **529,** Springer Verlag, New York, 1976.

[5] Hoversten, E.V., Snyder, D.L., Harger, R.O., and Kurimoto, K., "Direct-Detection Optical Communication Receivers", *IEEE Trans. Comm.,* Vol. **COMM-22,** No. 1, January 1974.

[6] Kailath, T. and Duttweiler, D., "An RKHS Approach to Detection and Estimation Problems - Part III", *IEEE Trans. Inf. Theory,* Vol. **IT-18,** No. 6, Nov.1972.

[7] Lazar, A.A. and Schwartz, S.C., "On Information Processing Using Counting Point Process Observations", Tech. Rept. No. 44, Information Sciences and Systems Laboratory, Dept. Electrical Eng. and Computer Science, Princeton University, Princeton, NJ 08544, Sept. 1979.

[8] Lazar, A.A., "Optimal Information Processing Using Point Process Observations", *Ph.D. dissertation,* Dept. Electrical Eng. and Computer Science, Princeton University, 1980.

[9] Liptser, R.S. and Shiryayev, A.N., *Statistics of Random Processes II: Applications,* Springer Verlag, New York, 1978.

[10] Macchi,O., "Distribution Statistique des Instants d'Émission des Photoélectrons d'une Lumière Thermique", *C.R. Acad. Sci.* (Paris), Vol. **272,** Ser. **A,** pp.437-440, 1971.

[11] Macchi, O. - Picinbono, B.C., "Estimation and Detection of Weak Optical Signals", *IEEE Trans. Inf. Theory,* Vol. **IT-22,** No. 5, Sept. 1972.

[12] Riesz, F. and Nagy, Sz., *Functional Analysis,* Ungar, New York, 1952.

[13] Snyder, D.L., "Filtering and Detection for Doubly Stochastic Poisson Processes", *IEEE Trans. Inf. Theory,* Vol. **IT-18,** No. 1, January 1972.

[14] -------, "Smoothing for Doubly Stochastic Poisson Processes", *IEEE Trans. Inf. Theory,* Vol. **IT-18,** No. 5, Sept.1972.

[15] Snyder, D.L. - Rhodes, I.B., "Filtering and Control Performance Bounds with Implications to Asymptotic Separation", *Automatica,* Vol. **8,** pp. 747-753, 1972.

[16] Snyder, D.L., *Random Point Processes,* Wiley Interscience, New York, 1975.

[17] Van Trees, H., *Detection, Estimation and Modulation Theory,* Vol. **III,** John Wiley, New York, 1971.

DETECTION OF SIGNALS WITH RANDOM PHASE IN SEA CLUTTER

A. H. EL-SAWY
Department of Electrical Communications and Electronics
Helwan University
Helwan, Egypt

ABSTRACT

Two methods for detection of signals with random phase in sea clutter are presented under the assumption that the interfering noise distribution is known only by quantiles. Using performance, simplicity of implementation and stability of the test thresholds as measures of superiority these two detectors are compared with other known detectors and are shown to be better.

I. Introduction

This paper investigates the problem of detection of signals with random phase in sea clutter. In such a case, the engineer faces the situation where the noise distribution is unknown and changes with sea state. Utilization of detectors which depends directly on the inphase and/or quadrature phase components as in [6], we should not expect good results, Since the distribution of these components will be the same except for a location shift, but this location shift will change randomly from one observation to another and will have a mean which is equal to zero under both hypotheses.

On the other hand, it was noticed that the mean of the envelope of the observations is monotonically increasing with the signal in the case of random phase [1 Figs. (4-7)]. As a result of the above observation, the main trend in treating the problem of detection of signals with random phase was generally to design detectors which use the envelope of the observations as input data instead of the components of the observations [2,3 and 4]. In addition, if we examine the results in [5 and 6] we may notice that, for the types of noise under considerations, censoring the observations always leads to improvement in the performance. This conclusion was also supported by some previous results by Trunk [2] when he tested the trimmed-mean detector.

As a result of the above discussion, we are going to consider in this paper only these detectors which utilize the envelope of the observations as input data, and which limit the effect of the tails of the observations distribution in some way. It should be emphasized here that we are looking for detectors which have good performance over a wide class of noise distributions when there is only parital information about these distributions. Also, we shall consider that the available information is always quantiles.

II. M-Detector on Envelope

Let $z_1, z_2, \ldots, z_N$ be independent identically distributed samples taken from the probability density $f(z)$ and let $F(z)$ be the distribution function of z. Consider also that

$$z_i = \sqrt{x_i^2 + y_i^2} \tag{1}$$

where

$$x_i = A \cos \phi_i + w_i \tag{2}$$

and

$$y_i = A \sin \phi_i + v_i \tag{3}$$

where v_i and w_i are noise components, ϕ_i is a random phase with uniform density function over the interval $[0,2\pi]$, and A is the signal amplitude which is equal to zero under H_0 and greater than zero under H_1. The exact value of A may be unknown. Assume also that there exist two known real numbers Z_1 and Z_2 such that $0<Z_1<Z_2<\infty$ and $F(Z_1|H_0)=P_1$ and $F(Z_2|H_0)=P_2$ where P_1 and P_2 are known and not equal to zero or one.

Before proceeding to describe the M-detector on envelope, we need the following definitions. Define the nonlinearity $l(\cdot)$ as

$$l(t) = \begin{cases} -b\,a & (t) \le Z_1 \\ -b\,a + \dfrac{(b+1)a}{(Z_2-Z_1)}(t-z_1) & Z_1 < t \le Z_2 \\ a & Z_2 < t \end{cases} \tag{4}$$

where a and b are two constants and Z_1, Z_2 are as defined before. Define θ_N as the value of θ such that

$$\sum_{i=1}^{N} g(z_i-\theta_N)=0 \tag{5}$$

$$g(z-\theta) = l(z-\theta) - E[l(z)|H_0] \tag{6}$$

Then the M-detector on envelope will be of the form

$$\sqrt{N}\ \theta_N \underset{H_0}{\overset{H_1}{\gtrless}} \gamma \tag{7}$$

where γ is the threshold.

Examining the above detection strategy, we find that the main idea is to use a fraction of the small observations (those less than Z_1) to get rid of the large observations (those larger than Z_2) without sorting the data. If b was chosen to be equal to $\frac{P_2}{P_1}$, then asymptotically this will be equivalent to the average sum of all observations in some intermediate interval of length (Z_2-Z_1). In this, the M-detector shares with the median detector the good property of deleting outlyers; however, it does not depend on one observation. So it is midway between the mean and median detectors.

To study the stability of the threshold we must first obtain an expression for the distribution of θ_N under H_0.

Lemma 1:

Under H_0, $\sqrt{N}\,\theta_N$ is asymptotically distributed as normal with zero mean and variance

$$V^2(\theta_N) = \frac{E[g^2(z)]}{[E'(g(z-\theta))|_{\theta=0}]^2} \tag{8}$$

To evaluate the sensitivity of the threshold with respect to the distribution, we calculate the explicit value of the variance in equation (8).

$$E'\{g(z-\theta)|_{\theta=0}\} = E'\{l(z-\theta)|_{\theta=0}\}$$

$$= \frac{(b+1)a}{(Z_2-Z_1)} \quad [P_2-P_1] \tag{9}$$

i.e. it is independent of the specific shape of $f(z)$.

$$E[g^2(z)] = E[l^2(z)] - E^2[l(z)]$$

$$= b^2a^2 P_1 + a^2[1-P_2] + [ba + \frac{(b+1)a}{Z_2-Z_1} Z_1]^2 [P_2-P_1][1+P_1-P_2]$$

$$+ [\frac{(b+1)a}{Z_2-Z_1}]^2 \{\int_{Z_1}^{Z_2} z^2 f(z)dz - [\int_{Z_1}^{Z_2} z\, f(z)dz]^2\}$$

$$- 2\ [ba + \frac{(b+1)a}{Z_2-Z_1} Z_1][\frac{(b+1)a}{Z_2-Z_1}]\ [1+P_1-P_2] \int_{Z_1}^{Z_2} z\, f(z)dz \tag{10}$$

It can be seen that the value of the threshold will change with changes in the true distribution only through the last two terms in equation (10). On the other hand, since we are dealing with detection in sea clutter, where all possible noise distributions range between Rayleigh and lognormal [7], a proper choice of Z_1 and Z_2 will minimize the effect of these two terms. For example, if we choose Z_1 and Z_2 such that P_1 and P_2 equal .1 and .5 respectively, the effect of these terms on the variance will be negligible. This choice also would help in calculating $E[l(z)|H_0]$ which is required to perform the test.

Receiver implementation

To perform the above test as it was described by equation (7), we must first calculate θ_N. This process may not be easy if the number of observations involved is large since $l(\cdot)$ is a nonlinear function. But, on the other hand, noting that $g(\cdot)$ is monotonic in its argument, for any number ε such that $\varepsilon > \theta_N$

$$\sum_{i=1}^{N} g(z_i-\varepsilon) < 0$$

and if $\varepsilon < \theta_N$, then

$$\sum_{i=1}^{N} g(z_i-\varepsilon) > 0$$

Accordingly if we determined the threshold to be γ, then the M-detector on envelope can be implemented in the form

$$\sum_{i=1}^{N} g(z_i-\gamma) \underset{H_0}{\overset{H_1}{\gtrless}} 0 \tag{11}$$

or, equivalently

$$\sum_{i=1}^{N} l(z_i-\gamma) \mathop{\gtrless}_{H_0}^{H_1} N\,E[l(z)|H_0] \tag{12}$$

The implementation of (12) is shown in figure (1).

III. Robust Quantizer Detectors

The main idea behind the set of detectors to be discussed in this section is in the utilization of a quantizer to transfer the input data to a set of preassigned values, then to utilize the quantizer outputs as an input to the decision rule. Assume that $\{z_i\}_{i=1}^{N}$ is the sequence of input data and $\{a_j\}_{j=0}^{m}$ is a sequence of real number such that $a_0=0$ and $a_m=\infty$. Then the output of the quantizer q_i will be given by

$$q_i = q(z_i) = l_j \qquad a_{j-1} \leq z_i < a_j \tag{13}$$

where $\{l_j\}_{j=1}^{m}$ is the set of preassigned values for the quantizer output. The test of hypotheses will be of the form

$$T_N = \frac{1}{N}\sum_{i=1}^{N} q_i \mathop{\gtrless}_{H_0}^{H_1} \gamma \tag{14}$$

where γ is the threshold. The above choice of T_N in fact does not in anyway limit the generality of the procedure since the q_i's are independent as long as the z_i's are, and we are still free to choose the l_j's to satisfy any optimality criterion we consider. Similar quantizer detectors were presented in [8,9] under the assumption that the noise distribution is known and in [10] under the assumption that the noise distribution is partially known. The following remarks are in order.

1) With the proper choice of $\{a_j\}$ the detector will limit the effect of large observations to the desired degree.
2) The detector is insensitive to the specific shape of the distribution.
3) If there is a family of distributions such that $[F(a_j)-F(a_{j-1})]$ under H_0 is the same for all members of this family for $j=1,\ldots, m$, then the distribution of T_N will be the same for any member of this family under H_0 and the test will be a constant false alarm rate test (CFAR) over this family.

From the above remarks and since we are concerned with the threshold stability, it seems that the most appropriate way to choose $\{a_j\}$ as in some way related to the quantiles of the distribution. In all detectors involved in this study, the a_j's were chosen such that $[F(a_j)-F(a_{j-1})] = \frac{1}{m}$ for all $j=1,\ldots, m$.

To choose the l_j's, two different methods were used

a) The fixed step quantizer

In this method, l_j was chosen as $l_j=j$. So, if we assume that n_j observations fall between $a_{(j-1)}$ and a_j, the test statistic can be rewritten as

$$T_N = \frac{1}{N}\sum_{j=1}^{m} j\, n_j \tag{15}$$

b) Coherent-Optimal quantizer:

In [10] it was shown that if the signal is a dc signal in additive noise then a locally most powerful quantizer detector of the above form can be constructed using

$$l_j = \frac{1}{m}\left[f_{(a_{j-1})} - f_{(a_j)}\right] \quad j=1,\ldots, m \tag{18}$$

Although this is not the case here, this quantizer was included in the study for the sake of completeness.

IV. The Binary Integrator Detector (BID)

This detector was presented under different names in [7,9,11 and 12]. The main idea behind this detector is to utilize two thresholds γ and K and to construct a test of the form

$$T_N = \sum_{i=1}^{N} U(Z_i - \gamma) \underset{H_0}{\overset{H_1}{\gtrless}} K \tag{19}$$

where $U(\cdot)$ is the unit step function and $\{Z_i\}$ is the sequence of observations.

The problem with such detectors is in the choice of the second threshold K. Test results show that the receiver performance depends on the choice of K and there is a different optimal choice for each distribution.

V. The Trimmed-Mean detector

In this section, we give a brief description of the trimmed-mean detectors which will be utilized in the next section for comparison. Assume there are N observations, then the designer chooses two numbers n_1 and n_2 such that $N \geq n_2 \geq n_1 \geq 0$, and the test statistics will be

$$T_N = \sum_{i=n_1}^{n_2} z_i \tag{20}$$

where z_i is the ith smallest observation, i.e. $z_1 \leq z_2 \leq \ldots \leq z_n$. One difficulty with this test is that it requires sorting the data. The second difficulty is in setting the threshold since it depends on the distribution and there is no mathematical expression to be used. Notice also that if $n_1 = n_2 = [\frac{N}{2}]$ this turns out to be the median detector and if $n_1 = 1$ and $n_2 = N$ it is the mean detector. On the other hand, it was shown in [2] that this detector performs better than most of the known detectors for both lognormal and contaminated normal distributions. More details about the trimmed-mean detector can be found in [13].

VI. Numerical Results

To examine the finite sample size performance of these detectors, simulation experiments were conducted at sample sizes of N=10 and 30 for the case when the noise has a lognormal distribution given by

$$f(z) = \frac{1}{\sqrt{2\pi z}} \, e^{\frac{-\ln^2 z}{2\sigma^2}} \tag{21}$$

where $\sigma = 6$ db, and at N=10 for the contaminated normal distribution given by

$$f(z) = (1-r^2)\, e^{\frac{-z^2}{2}} + \frac{r^2 z^2}{k^2}\, e^{\frac{-z^2}{2k^2}} + \frac{2\,(1-r)\,z}{k}\, \exp\left[\frac{-z^2(k^2+1)}{4k^2}\right] I_0\left[\frac{z^2(k^2-1)}{4k^2}\right] \tag{22}$$

where $I_0(\cdot)$ is the modified Bessel function, r=0.25 and k=2.25.

For the M-detector on envelope, the values of Z_1 and Z_2 were found such that P_1 and P_2 equal 0.1 and 0.5 respectively. b was taken to be $\frac{P_2}{P_1} = 5$. To calculate $E[l(z)|H_0]$, a Rayleigh approximation of the form

$$f(z) = \frac{(z-m)}{\sigma^2}\, e^{\frac{-(z-m)^2}{2\sigma^2}} \tag{23}$$

was utilized. First, we calculate m and σ such that this distribution satisfies the requirements at Z_1 and Z_2. Then, we use this approximation to calculate $E[l(z)|H_0]$. It was found that the maximum difference between the approximate and true values is about 5%. It was noticed also that the value of m is very small in both cases.

Figures (2-8) give the results of these simulations. The signal to noise ratio was taken as $20 \log_{10}(A/M)$ where A is the signal amplitude and M is the median of the distribution under consideration. The performance of the fixed step quantizer was much better than the optimal-coherent quantizer so the results of the second will not be reported here. The following remarks can be made about the M-detector:

1) $\frac{Z_1}{Z_2}$ is almost the same for the two distributions.
2) At N=10, if the thresholds were normalized by the appropriate value of Z_2, then the thresholds which would give a P_F of 10^{-2} and 10^{-6} when the noise is lognormal, would give a P_F of 10^{-2} and 10^{-7} when the noise is contaminated normal. This confirms the stability of the threshold.
3) At N=10, using the same normalization we found that at the same signal to noise ratio the probability of detection is also almost constant, independent of the true distribution.

Remarks 2 and 3 imply that the detector is robust or insensitive to changes in distribution, even at small sample sizes such as N=10. More about this detector will be presented later in this chapter.

For the case of trimmed-mean M_1 and M_2 was taken to be equal to 2 and 5 respectively since this insures better results [13].

VII. Detector Comparison

a) Performance

Comparing the performance results of the chosen detectors, it should be noted that for the lognormal distribution at N=10 the performance of the M-detector on envelope, the trimmed-mean detector (n_1=2, n_2=5) and the fixed step quantizer detector are all very comparable. In fact the graphs for P_d against (S/N) overlap in most cases, and all of them are within less then 1db. Also, the performance of the M-detector gets better than the others as P_F decreases and P_d increases. The B.I.D. detector always has poorer performance than the others, even for K=7 which was shown to be optimal for lognormal when N=10 [7]. Figure (9) shows an example of these comparisons at $P_F=10^{-6}$. It shows also the Chernoff bound as an approximation for the performance of the optimal detector. Both the M-detector and the fixed quantizer detectors are the closest to this bound, with the M-detector better at higher values of P_d.

At N=30, the performance of both the M-detector and the fixed step quantizer is almost the same.

For the contaminated normal, the performance of the trimmed mean and M-detector was the same as mentioned before. The B.I.D. was the worst detector in the group. The fixed quantizer is very slightly better than the M-detector at $P_F=10^{-2}$ but the M-detector is much better at $P_F \leq 10^{-4}$.

b) General Properties

1) The performance of the B.I.D. detector is highly dependent on the value of the second threshold K and the true noise distribution, and since the main reason behind this study is the uncertainty about the noise distribution, this detector cannot be recommended for applications.

2) Although the trimmed-mean detector has comparable performance to the other two good detectors, implementation seems impractical since it requires sorting the data, and there is no known way to set the threshold for it.

3) The M-detector and the fixed step quantizer detector are the simplest to implement among all detectors studied, and they also have the best performance.

4) The M-detector has the advantage over the fixed step quantizer detector that it requires less information. It requires knowledge of only two quantiles, while the quantizer detector requires more quantiles.

5) Using the quantizer detector, P_F will be set only approximately and some values of P_F are unachievable for finite N.

6) If the designer is interested in the low P_F - high P_d region, the M-detector has better performance than the quantizer detector.

VIII. Conclusion

The problem of detection of signals with random phase in sea clutter-like noise was treated in this paper. Several detectors were developed and/or described. Comparing these detectors as to performance and implementation it was found that the M-detector on envelope is best.

REFERENCES

[1] S. F. George, "The Detection of Nonfluctuating Targets in Log-Normal Clutter," NRL Rep. 6796, Oct. 1968.

[2] G. V. Trunk, "Non-Rayleigh Sea Clutter: Properties and Detection of Targets," NRL Rep. 7986, June 1976.

[3] A. D. Whalen, "Detection of Signals in Noise," Acad. Press, NY, 1971.

[4] G. V. Trunk, "Median Detector for Noncoherent Distributions," NRL Rep. 6898, May 1969.

[5] A. H. El-Sawy and V. D. VandeLinde, "Robust Detection of Known Signals," IEEE Trans. Inform. Theory, Vol. IT-23, pp. 722-727, Nov. 1977.

[6] A. H. El-Sawy, "Detection of Signals with unknown phase,"same proceedings.

[7] D. Schleher, "Radar Detection in Log-Normal Clutter," Ph.D. Dissertation, Polytech. Inst. of NY, June 1975.

[8] S. A. Kassam, "Optimum Quantization for Signal Detection," IEEE Trans. Comm., vol. COM-25, pp. 479-484, May 1977.

[9] H. V. Poor and J. B. Thomas, "Maximum-Distance Quantization for Detection," Proceedings of the Fourteenth Annual Allerton Conf. on Circuit and System Theory, pp. 925-934, Sep. 1976.

[10] Y. Ching and L. Kurz, "Nonparametric Detectors Based on m-Interval Partitioning," IEEE Trans.Inform.Theory,Vol.IT-18, pp.251-257,Mar 1972.

11 L. B. Milstein, et al., "Robust Detection Using Extreme-Value Theory," IEEE Trans. Inform. Theory, vol. IT-15, May 1969.
12 M. Schwartz, "A Coincidence Procedure for Signal Detection," IEEE Trans. Inform. Theory, vol. IT-11, Jan. 1977.
13 G. V. Trunk, "Trimmed-Mean Detector for Noncoherent Distributions," NRL Rep. 6997, Dec. 1969.

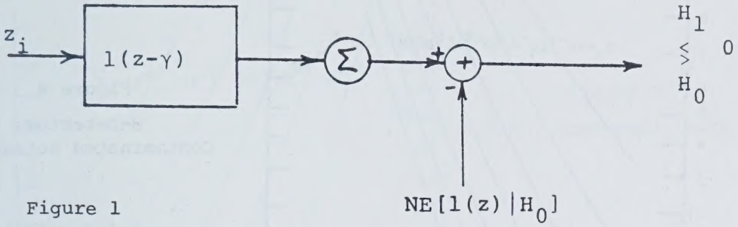

z_i
$l(z-\gamma)$
Σ
+
-
$\overset{H_1}{\underset{H_0}{\lessgtr}}\ 0$
$NE[l(z)|H_0]$

Figure 1

M-Detector on envelope

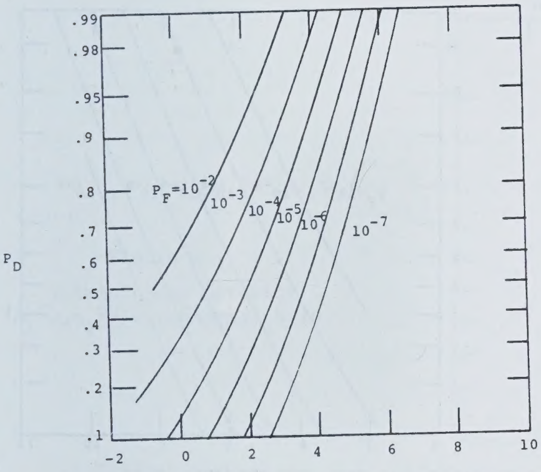


Figure 2

M-Detector, Lognormal noise, N=30

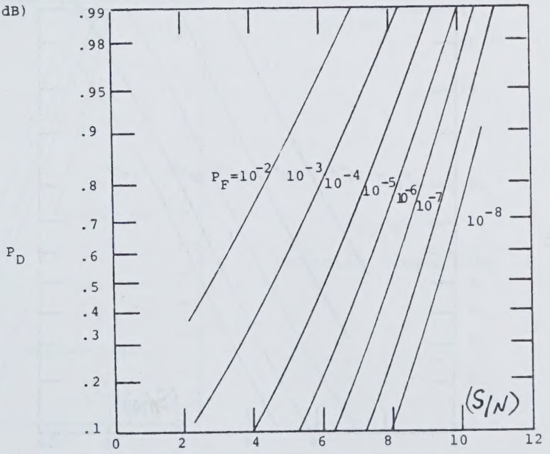


Figure 3

M-Detector, Lognormal noise, N=10

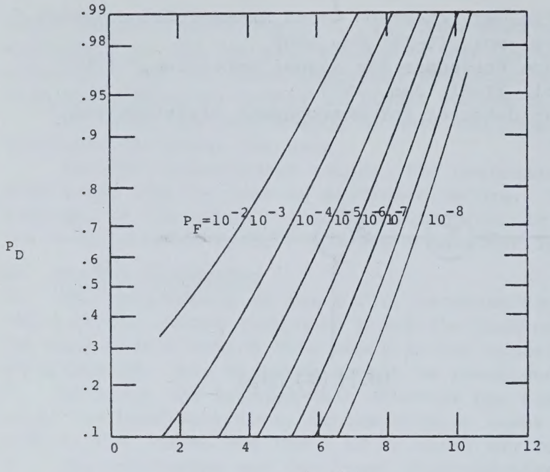


Figure 4

M-Detector,
Contaminated Noise, N=10

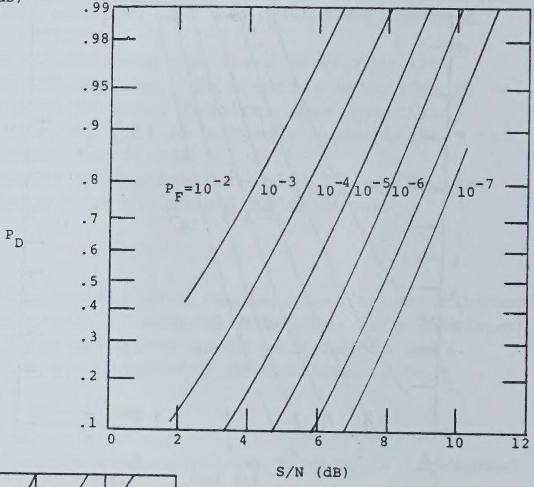


Figure 5

Equally spaced quantizer,
lognormal, N=10

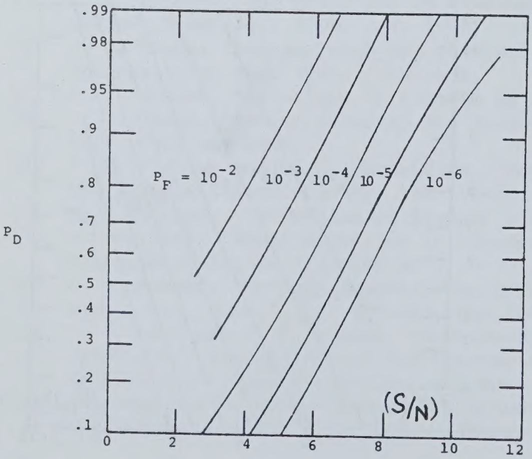


Figure 6

Equally spaced quantizer,
contaminated normal, N=10

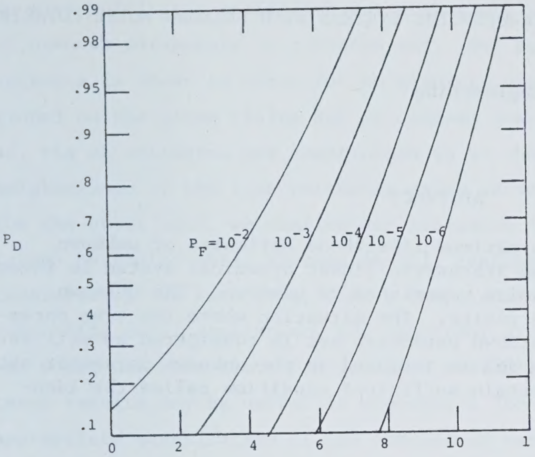


Figure 7

B.I.D., Lognormal, N=10

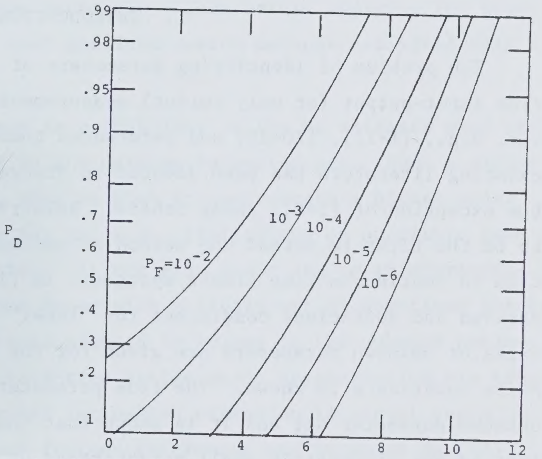


Figure 8

Trimmed Mean Detector

$(n_1=2, n_2=5)$, lognormal N=10

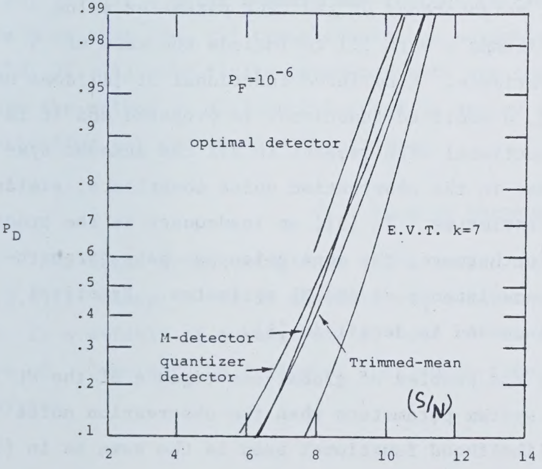


Figure 9

Detector Comparison

GLOBAL IDENTIFICATION OF CONTINUOUS-TIME SYSTEMS WITH UNKNOWN NOISE COVARIANCE

JITENDRA K. TUGNAIT
Division of Information Engineering
The University of Iowa
Iowa City, Iowa, 52242

ABSTRACT

Global convergence of the maximum likelihood estimates of unknown parameters of a continuous-time stochastic linear dynamical system is investigated when the observation noise covariance is unknown. The unknown parameter set is assumed to be finite. The situation where the true parameter does not belong to the unknown parameter set is considered as well as the situation where the true model is included in the unknown parameter set. Convergence is proved under certain sufficient condition called the Identifiability Condition.

I. INTRODUCTION

The problem of identifying parameters of stochastic dynamical systems from input-output (or only output) measurements has been studied extensively; see, e.g., [1-11], [20-25] and references therein. However, most of the existing literature has been devoted to discrete-time dynamical systems, with the exception of [1-6], among others. Balakrishnan [1, Chapter 8] appears to be the first to extend the method of maximum likelihood (ML) identification to continuous-time linear systems. In [1] a likelihood functional is derived and sufficient conditions for "local" consistency of the ML estimates of unknown parameters are given for the case when the observation noise covariance is known. The true parameter is assumed to belong to the unknown parameter set and it is shown that the ML estimate is weakly consistent in any arbitrarily small neighborhood of the true parameter value. Bagchi [3, 4] extended Balakrishnan's work [1] to include the case of unknown measurement noise covariance. Likelihood functional of [1] does not exist in this case; in [3, 4], a modified functional is proposed and it is shown that maximizing this functional with respect to all the unknown system parameters, including those in the observation noise covariance, yields locally consistent parameter estimates. In [5], an inadequacy in the proof of [1] has been removed and furthermore, the conclusion has been strengthened by proving strong local consistency of the ML estimates. Practical aspects of the problem are discussed in detail in [6].

In this note we consider the problem of global convergence of the ML estimates of all the unknown system parameters when the observation noise covariance is unknown. The likelihood functional used is the same as in [3].

We give sufficient conditions for weak global consistency of the ML estimates of unknown parameters on a finite set. Our results are global in that convergence is shown to occur for an absolute maximum of the likelihood functional on the given finite set of unknown parameter values, whereas in [3, 4], the ML estimates are constrained to be chosen from a sufficiently small neighborhood of the true parameter value in order for convergence to occur. (On the other hand, we confine our attention to finite (albeit arbitrarily large) parameter sets, whereas in [1], continuous parameter uncertainty is considered.) We note that, from a computational viewpoint, results of [3, 4] concerning local consistency are important since to obtain the ML estimate one would use some sort of a gradient search technqiue. Our global consistency results may be useful in choosing a "coarse" parameter estimate (after appropriate quantization of the continuous parameter space, if necessary) to narrow down the range of unknown parameter values before applying the gradient search techniques. Note that gradient search methods can yield only a local extremum.

We also analyze convergence in probability of the ML estimate when the true parameter does not belong to the unknown parameter set. Such a situation could arise, for example, when we wish to approximate a higher order system with lower order models due to, e.g., limitations on available computation and storage capabilities. It can also occur due to an approximation of the continuous parameter space with a finite set of quantized points. In [1, 3-5], the true parameter is assumed to belong to the unknown parameter set. The problem of discrete-time system model approximation via identification techniques has received increasing attention in recent years [20, 22-25]; a brief summary and further references may be found in [23]. We note that [24, 25] treat continuous parameter uncertainty, whereas in [20, 22, 23] only finite parameter sets are considered. In this note, we use the method of ML identification to arrive at a criterion for model selection for continuous-time systems.

II. PROBLEM STATEMENT

We begin by assuming that there exists an underlying probability space $(\Omega, \mathcal{B}, P)$ where Ω is an abstract set consisting of elementary events ω, $\mathcal{B}$ is a σ-field of subsets of Ω, and P is a probability measure defined on $\mathcal{B}$. Consider the following continuous-time stochastic linear dynamical system

$$x(t;\omega) = \int_0^t Ax(s;\omega)ds + \int_0^t Bu(s)ds + \int_0^t F\, dW_1(s;\omega) + x(0;\omega) \quad (1)$$

$$y(t;\omega) = \int_0^t C\, x(s;\omega)ds + \int_0^t Du(s)ds + \int_0^t G\, dW_2(s;\omega) \quad (2)$$

where $u(t)$ is a p-dimensional known input function; $x(t;\omega)$ and $y(t;\omega)$ are n- and m-dimensional state and observation vectors, respectively; $A, B, C, D, F,$ and G are $n \times n$, $n \times p$, $m \times n$, $m \times p$, $n \times n$, and $m \times m$ constant matrices, respectively; $W_1(t:\omega)$ and $W_2(t;\omega)$ are n- and m-dimensional independent Wiener processes, respectively, with identity covariances. Without loss of generality, we shall take $x(0;\omega) = 0$ since we are interested in asymptotic properties of the system (see also condition (C1) in the sequel). Furthermore, we take $W_1(0;\omega) = W_2(0;\omega) = 0$. Finally, we assume that $GG^* > 0$[†] to exclude the singular case.

The identification problem is that of estimating unknown parameters in the matrices $A, B, C, D, F,$ and G, given $y(t;\omega)$ and $u(t)$, $0 \leq t \leq T$, for one realization of the experiment.

Let θ denote the vector of all the unknown system parameters and let θ_0 be the true value. It is assumed that $\theta \in \Theta$, where Θ contains only a finite number of elements. To each $\theta \in \Theta \cup \{\theta_0\}$ there corresponds a probability measure $p_{y,\theta}$ on $\not{C}$ induced by the process $y(t;\omega)$, $0 \leq t < \infty$, assuming that $y(t;\omega)$ results from (1) - (2) with θ as the "true" parameter; here $\not{C} = C^m[0,t]$ denotes the space of continuous functions from $[0,T]$ into R^m. Let $\hat{\theta}_T(\omega)$ denote the ML estimate of θ on Θ based on the input-output measurements $u(t)$ and $y(t;\omega)$, $0 \leq t \leq T$; an ML estimate on Θ is an estimate $\hat{\theta}_T(\omega) \in \Theta$ such that

$$H(\hat{\theta}_T(\omega);y(\cdot;\omega);T) \geq H(\theta;y(\cdot;\omega);T) \quad (\forall\theta \in \Theta)$$

where $H(\theta;y(\cdot;\omega);T)$ is a likelihood functional to be defined later. Our main objective is to analyze the asymptotic behavior of $\hat{\theta}_T(\omega)$ as $T \to \infty$ for both the cases, when $\theta_0 \in \Theta$ and when $\theta_0 \notin \Theta$. The situation $\theta_0 \notin \Theta$ implies that the true probability measure p_{y,θ_0} is not in the set parameterized by θ, $\{p_{y,\theta} : \theta \in \Theta\}$. In particular, we investigate sufficient conditions for convergence in probability[†] (i.p.) of the ML estimate of θ on Θ to θ_0 when $\theta_0 \in \Theta$ and to a set in Θ when $\theta_0 \notin \Theta$.

[†] G^* denotes transpose of G and $GG^* > 0$ denotes that the matrix GG^* is positive definite.

III. LIKELIHOOD FUNCTIONAL

We shall use the modified likelihood functional proposed by Bagchi [3, 4]. It is given by

$$H(\theta;y(\cdot;\omega);T) = \exp\left\{-\frac{T}{2}\, q[\theta;y(\cdot;\omega);T]\right\} \tag{3}$$

where

$$q[\theta;y(\cdot;\omega);T] = \left[\frac{1}{T}\left(\int_0^T \left[\left(\tilde{G}_0\tilde{G}_0^*\right)^{-1} (D_\theta u(t) + C_\theta \hat{x}(t;\theta;\omega)),\ D_\theta u(t) + C_\theta \hat{x}(t;\theta;\omega)\right] dt \right.\right.$$
$$\left.\left. -\,2\int_0^T \left[\left(\tilde{G}_0\tilde{G}_0^*\right)^{-1} (D_\theta u(t) + C_\theta \hat{x}(t;\theta;\omega)),\ dy(t;\omega)\right]\right)\right] \tag{4}$$

and where $[\cdot,\cdot]$ denotes inner product in R^m and the second integral is to be interpreted in the Itô sense. Note that we indicate the dependence of various quantities in (4) on θ explicitly. Furthermore, $\hat{x}(t;\theta;\omega)$ satisfies the stochastic differential equation

$$\hat{x}(t;\theta;\omega) = \int_0^t A_\theta \hat{x}(s;\theta;\omega)ds + \int_0^t B_\theta u(s)ds + \int_0^t P_\theta(s)C_\theta^*\left(G_\theta G_\theta^*\right)^{-1} dz(s;\theta;\omega) \tag{5}$$

$$z(t;\theta;\omega) = y(t;\omega) - \int_0^t C_\theta \hat{x}(s;\theta;\omega)ds - \int_0^t D_\theta u(s)ds \tag{6}$$

where $P_\theta(t)$ satisfies the matrix Riccati equation

$$dP_\theta(t)/dt = A_\theta P_\theta(t) + P_\theta(t)A_\theta^* + F_\theta F_\theta^* - P_\theta(t)C_\theta^*\left(G_\theta G_\theta^*\right)^{-1} C_\theta P_\theta(t)\ ;$$
$$P(0) = 0. \tag{7}$$

Here $\hat{x}(t;\theta;\omega)$ is the MMSE estimate of $x(t;\omega)$ based on the observation sample $y(\tau;\omega)$, $0 \le \tau \le t$ assuming that the system (1) - (2) generates the given $y(\tau;\omega)$, $0 \le \tau \le t$ with θ as the "true parameter, i.e., $\hat{x}(t;\theta;\omega) = E\{x(t;\omega)|\mathcal{B}_y(t),\theta\}$ where $\mathcal{B}_y(t)$ denotes the smallest σ-algebra generated by $y(\tau;\omega)$, $0 \le \tau \le t$. Furthermore, $P_\theta(t)$ is the corresponding state estimation error covariance, and $z(t;\theta;\omega)$ is a Wiener process, the so-called innovations process, for $\theta = \theta_0$, the true parameter value, with

---

[†] If not explicitly indicated, the probability measure used in P_{y,θ_0} which is the one corresponding to (or indexed by) the true parameter θ_0.

$$E\{z(t;\theta_0;\omega)z^*(s;\theta_0;\omega)\} = G_{\theta_0} G^*_{\theta_0} \min(t,s) \quad . \tag{8}$$

Finally, $\tilde{G}_0$ is an arbitrary "guess" of G_{θ_0}, the true value of G, such that $\tilde{G}_0\tilde{G}^*_0 > 0$ [3, 4].

In Refs. [3, 4] it has been shown that minimizing the functional $q[\cdot]$ in any arbitrarily small neighborhood of θ_0 will yield locally consistent estimates of all the unknown parameters of the system under a certain sufficiency condition. Our main concern in this note is to investigate global convergence of the ML estimate of θ on Θ. In other words, we investigate the asymptotic behavior of the absolute minimum of q (or absolute maximum of H) on Θ.

Since we are interested in the asymptotic properties of the estimates, conditions (C1) - (C3) are imposed on the system (1) - (2) to ensure convergence of certain expressions as t becomes very large [1, 3, 4].

(C1) For any θ in $\Theta \cup \{\theta_0\}$, A_θ is a stable matrix, i.e., all the eigenvalues of A_θ have negative real parts.

(C2) For any θ in $\Theta \cup \{\theta_0\}$, the pair (C_θ, A_θ) is completely observable, that is, rank $\left[C^*_\theta | A^*_\theta C^*_\theta \mid \cdots \mid \left(A^*_\theta\right)^{n-1} C^*_\theta\right] = n$.

(C3) Assume that the known input $u(\cdot)$ is such that

$$\lim_{T\to\infty} \frac{1}{T}\int_0^T ||u(t)||^2 \, dt < \infty \qquad \text{(exists and is finite)}$$

and

$$r_u(t) = \lim_{T\to\infty} \frac{1}{T}\int_0^T u(s)u^*(s+t)ds$$

is a continuous function of t in every finite interval. Moreover, $\sup_t ||u(t)|| < \infty$.

Condition (C1) implies that the initial condition of the state has no effect on the asymptotic behavior of the system and this justifies setting the initial condition to be zero in (1). Condition (C2) implies that $\lim_{t\to\infty} P_\theta(t)$ exists and is unique; we denote this limit by P_θ which satisfies the algebraic equation

$$A_\theta P_\theta + P_\theta A^*_\theta + F_\theta F^*_\theta - P_\theta C^*_\theta \left(G_\theta G^*_\theta\right)^{-1} C_\theta P_\theta = 0 \tag{9}$$

and furthermore, the matrix $A_\theta - P_\theta C^*_\theta \left(G_\theta G^*_\theta\right)^{-1} C_\theta$ is stable.

Let $q(\theta;T) = E\{q[\theta;y(\cdot;\omega);T]\}$ and define $q(\theta) = \lim_{T\to\infty} q(\theta;T)$. In [4], following [1] (see also [3]), it has been shown that $q(\theta)$ exists.

Lemma 1, proved in [3, Lemma 3], [4, Lemma 3.4] following [1], is central to development of results in the sequel.

Lemma 1: Assuming that conditions (C1) - (C3) are satisfied,

$$\lim_{T\to\infty} E\{|q[\theta;y(\cdot;\omega);T] - q(\theta)|^2\} = 0$$

uniformly with respect to θ in compact subsets.

In the sequel, we use Lemma 1 to investigate convergence of the ML estimate of θ on Θ as $T \to \infty$.

IV. CONVERGENCE OF ML ESTIMATE

In this section we investigate sufficient conditions for global convergence (in probability) of the ML estimate of θ on Θ. We consider both the cases: when $\theta_0 \in \Theta$ and when $\theta_0 \notin \Theta$.

Theorem 1 specifies the "limit set" of the ML estimate as $T \to \infty$.

Theorem 1: Assuming that conditions (C1) - (C3) are satisfied, the ML estimate of θ on Θ belongs to a set Θ_ℓ in probability as $T \to \infty$ where

$$\Theta_\ell = \{\theta : \theta \in \Theta,\ q(\theta) = \min_{\theta\in\Theta} q(\theta)\}$$

Remark 1: Theorem 1 does not imply convergence of the ML estimate of θ to some θ' in Θ_ℓ. It is possible that the ML estimate might "oscillate" among various θ' in Θ_ℓ.

We now turn to the question whether the set Θ_ℓ contains the true parameter θ_0 when $\theta_0 \in \Theta$. Lemma 2 shows that $q(\theta) \geq q(\theta_0)$; therefore, $\theta_0 \in \Theta_\ell$ whenever $\theta_0 \in \Theta$.

Lemma 2: Assuming that conditions (C1) - (C3) are satisfied, $q(\theta) \geq q(\theta_0)$ $\forall\theta \in \Theta$.

In light of Lemma 2, Theorem 1 is now modified as follows for the case $\theta_0 \in \Theta$.

Theorem 2 Assuming that conditions (C1) - (C3) are satisfied and $\theta_0 \in \Theta$, the ML estimate of θ on Θ belongs to a set Θ_ℓ^0 in probability as $T \to \infty$ where

$$\Theta_\ell^0 = \{\theta : \theta \in \Theta \quad \text{and} \quad q(\theta) = q(\theta_0)\} \quad .$$

We shall now consider the special case when, in (1) - (2), $u(t) \equiv 0$.

Lemma 3: Let $u(t) \equiv 0$ in (1) - (2) and assume that conditions (C1) - (C2) are satisfied. Then

$$q(\theta) - q(\theta_0) = \mathrm{tr}\, G_s \left[P_1^0(\theta) - P_1^0(\theta_0)\right] G_s$$

where $G_s G_s = \tilde{G}_0 \tilde{G}_0^{*\,-1}$, and $G_s > 0$,

$$P_1^0(\theta) \triangleq \lim_{T\to\infty} P_1^0(t;\theta)$$

$$P_1^0(t;\theta) \triangleq E\{[x_1(t;\omega) - \hat{x}_1(t;\theta;\omega)][x_1(t;\omega) - \hat{x}_1(t;\theta;\omega)]^*\}$$

$$x_1(t;\omega) \equiv C_{\theta_0}\ x(t;\omega) + D_{\theta_0}\ u(t)$$

and $x(t;\omega)$ is given by (1) with $A \equiv A_{\theta_0}$ etc.

Now we seek to show that $\lim_{T\to\infty} P_1^0(t;\theta)$ is indeed defined $\forall \theta \in \Theta$. From (1), (2), (5), and (6), we have

$$\begin{bmatrix} x(t;\omega) \\ \hat{x}(t;\theta;\omega) \end{bmatrix} = \int_0^t A'_\theta \begin{bmatrix} x(s;\omega) \\ \hat{x}(s;\theta;\omega) \end{bmatrix} ds + \int_0^t F'_\theta \begin{bmatrix} dW_1(s;\omega) \\ dW_2(s;\omega) \end{bmatrix} \tag{10}$$

where

$$A'_\theta \equiv \begin{bmatrix} A_{\theta_0} & 0 \\ P_\theta C_\theta^* G_\theta G_\theta^{*\,-1} C_{\theta_0} & A_\theta - P_\theta C_\theta^* G_\theta G_\theta^{*\,-1} C_\theta \end{bmatrix}$$

$$F'_\theta \equiv \begin{bmatrix} F_{\theta_0} & 0 \\ 0 & P_\theta C_\theta^* G_\theta G_\theta^{*\,-1} G_{\theta_0} \end{bmatrix}$$

and where we have used the steady-state version of the Kalman-Bucy filter (5) - (6) so that P_θ is given by (9); there is not loss of generality in this, however, algebraic manipulations are considerably reduced. Now let

$$L(t;\theta;\theta_0) \triangleq E\left\{\begin{bmatrix} x(t;\omega) \\ \hat{x}(t;\theta;\omega) \end{bmatrix} [x^*(t;\omega) \quad \hat{x}^*(t;\theta;\omega)]\right\} . \tag{11}$$

Then from (10) - (11) we have [1, Equation (2-13)]

$$dL(t;\theta;\theta_0)/dt = A_\theta' L(t;\theta;\theta_0) + L(t;\theta;\theta_0)A_\theta'^* + F_\theta' F_\theta'^* . \tag{12}$$

Now $L(\theta;\theta_0) \triangleq \lim_{t\to\infty} L(t;\theta;\theta_0)$ exists and is finite (irrespective of initial condition) if A_θ' is stable (i.e., if all eigenvalues of A_θ' have negative real parts) [1, p. 144]; this is indeed true $\forall\theta \in \Theta$ because of conditions (C1) - (C2). Therefore, $L(\theta;\theta_0)$ satisfies the algebraic equation

$$A_\theta' L(\theta;\theta_0) + L(\theta;\theta_0)A_\theta'^* + F_\theta' F_\theta'^* = 0 . \tag{13}$$

Finally

$$\lim_{t\to\infty} P_1^0(t;\theta) \triangleq P_1^0(\theta) = C_\theta' L(\theta;\theta_0) C_\theta'^* \tag{14}$$

where

$$C_\theta' = \begin{bmatrix} C_{\theta_0} & -C_\theta \end{bmatrix} .$$

For $\theta = \theta_0$, we have $P_1^0(\theta_0) = C_{\theta_0} P_{\theta_0} C^*_{\theta_0}$.

Remark 2: Any sufficient condition ensuring consistency of the estimate may be called an "Identifiability Condition." From Theorem 2, a sufficient condition is provided by the inequality

$$q(\theta) \neq q(\theta_0) \qquad (\forall\theta \in \Theta) \qquad (\theta \neq \theta_0) . \tag{15}$$

However, since θ_0 is not known, a verifiable Identifiability Condition would be to require (15) to hold for all pairs of θ's in Θ. We also note that our Identifiability Condition is weaker than the conditions of [1, 3, 4] in that it is implied by the conditions of [1, 3, 4]. Finally, consider the special case when, in (1) - (2), $u(t) \equiv 0$. In this case, from Lemma 3, the Identifiability Condition (15) becomes

$$\mathrm{tr}\, P_1^0(\theta) \neq \mathrm{tr}\, P_1^0(\theta_0) \qquad (\forall\theta \in \Theta) \qquad (\theta \neq \theta_0) . \tag{16}$$

Model Approximation via ML Identification

We now focus attention on the case when $\theta_0 \notin \Theta$. There are several cases of interest where such a situation arises. Suppose a global maximum likelihood parameter estimate under continuous parameter uncertainty is desired. Since the gradient search techniques can only find a local extremum, a two-step parameter estimation procedure has been suggested [20, 22]. As a first step, one obtains a "coarse" parameter estimate on a finite parameter set which is an appropriate approximation to the (original) continuous (infinite) uncertain parameter set; then, in the second stage, gradient search techniques may be used to "refine" the parameter estimate in some neighborhood of the coarse estimate. The true parameter may not belong to the finite parameter set after quantization of the continuous uncertain parameter set. Another situation of interest arises when one wishes to approximate a higher order system with a lower order model due to e.g., limitations on the available computation and storage capabilities. We use maximum likelihood as a criterion for choosing a model from Θ which best fits the model represented by θ_0, given input-output measurements $u(t)$ and $y(t;\omega)$, $0 \leq t \leq T$.

Theorem 1 defines the limit set of the ML estimate of θ on Θ as $T \to \infty$. The limit set Θ_ℓ may be redefined as follows

$$\Theta_\ell = \left\{ \theta : \arg\left(\min_{\theta\epsilon\Theta} q(\theta) - q(\theta_0) \right) \right\} .$$

Therefore, the ML identification technique leads to the selection of a model $\theta \in \Theta$ which asymptotically minimizes the difference $q(\theta) - q(\theta_0)$.

Remark 3: It should be pointed out that (particularly in case of the model approximation problem) various θ's in Θ may correspond to matrices A, B, C, D, F, and G which are of different dimensions (possibly lower) than the "true" matrices corresponding to θ_0. It is a tacit assumption in this paper that conditions (C1) - (C3) hold in this case too with obvious modifications; e.g., in (C2), rank of the observability matrix is "full" (which may not be n).

V. CONCLUDING REMARKS

The problem of global convergence of the maximum likelihood estimates of unknown parameters of a continuous-time stochastic linear system has been investigated when the observation noise covariance is unknown. The unknown parameter set was assumed to be finite. Convergence was proved under certain sufficient condition called the Identifiability Condition. As in [3], we considered convergence in probability; however, we considered absolute maximum of the likelihood functional, whereas in [3], consistency of a "local" maximum in a sufficiently small neighborhood of the true parameter value was proved. Moreover, we also investigated the case when the true model is not included in the unknown parameter set.

REFERENCES

[1] A. V. Balakrishnan, Stochastic Differential Systems I. Berlin: Spring-Verlag, 1973.

[2] A. V. Balakrishnan, J. Opt. Theory Appl., 9, pp. 187-212, 1972.

[3] A. Bagchi, Automatica, 11, pp. 533-536, Sept. 1975.

[4] A. Bagchi, Ph.D. Thesis in Engr., Univ. of Calif., Los Angeles, 1975.

[5] A. Bagchi, IMA Conf., Oxford, England, Sept. 1978.

[6] K. W. Iliff and R. E. Maine, J. of Guidance & Control, 2, pp. 228-234, May-June, 1979.

[7] E. Tse and J. J. Anton, IEEE Trans. Automatic Control, AC-17, pp. 637-646, Oct. 1972.

[8] K. J. Åström and P. Eykoff, Automatica, 7, pp. 123-162, 1971.

[9] R. K. Mehra and D. G. Lainiotis (eds.), System Identification: Advances and Case Studies. New York: Academic Press, 1976.

[10] G. C. Goodwin and R. L. Payne, Dynamic System Identification: Experiment Design and Data Analysis. New York: Academic Press, 1977.

[11] Special Issue, IEEE Trans. Automatic Control, AC-19, Dec. 1974.

[12] R. B. Ash, Real Analysis and Probability. New York: Academic Press, 1972.

[13] R. Bellman, Introduction to Matrix Analysis, New York: McGraw-Hill, 1960.

[14] J. L. Doob, Stochastic Processes, New York: Wiley, 1953.

[15] E. Wong, Stochastic Processes in Information and Dynamical Systems, New York: McGraw-Hill, 1971.

[16] S. Kullback, Information Theory and Statistics, New York: Wiley, 1959.

[17] H. Cramer, Mathematical Methods of Statistics, Princeton, N.J.: Princeton Univ. Press, 1946.

[18] A. Wald, Annals Math. Statistics, Vol. 20, pp. 595-601, 1949.

[19] J. Wolfowitz, Annals Math. Statistics, Vol. 20, pp. 601-602, 1949.

[20] Y. Baram and N. R. Sandell, Jr., IEEE Trans. Automatic Control, AC-23, pp. 61-66, Feb. 1978.

[21] Y. Baram, IEEE Trans. Automatic Control, AC-23, pp. 958-960, Oct. 1978.

[22] R. M. Hawkes and J. B. Moore, Proc. IEEE, 64, pp. 1143-1150, Aug. 1976.

[23] B. D. O. Anderson, J. B. Moore, and R. M. Hawkes, Automatica, 14, pp. 615-622, Nov. 1978.

[24] L. Ljung, in System Identification: Advances and Case Studies, R. K. Mehra and D. G. Lainiotis, ed., New York: Academic, 1976.

[25] P. E. Caines, IEEE Trans. Automatic Control, AC-21, pp. 500-505, Aug. 1976.

MEDIAN FILTERING

V.K. AATRE, E. ATAMAN AND K.M. WONG
Nova Scotia Technical College
Department of Electrical Engineering
P.O. Box 1000, Halifax, N.S. B3J 2X4

ABSTRACT

This paper discusses a nonlinear filtering technique, namely median filtering and its applications to speech and picture processing. The main advantage of the Median filtering is the preservation of sharp discontinuities and boundaries.

I Introduction

A problem often encountered in signal processing applications is the separation of the input into wanted and unwanted bundles. More often than not these two components of the input have different, and usually disjoint, spectra. To separate and to suppress the unwanted part of the input one often uses spectrum shaping filters. A broad class of filters that have found ever increasing use are the linear filters. For example, a low-pass filter is used to recover the signal from a noisy input where the spectrum of the noise occupies a band of frequencies higher than the signal frequencies. A main property of linear filters is the superposition property. The design of such filters have received considerable attention of many researchers.[1,2]

Not withstanding the wide application of linear filters, there are situations when linear filtering is inadequate. If the signal displays sharp discontinuities (indicating the presence of high frequency components) in addition to being corrupted by high frequency noise, then a linear filter designed to eliminate the noise will also smooth out the signal. Fig. 1 shows such a situation where (a), (b) and (c) are the actual signal, signal plus noise, and the linearly smoothed output, respectively. In such a situation one needs to take a recourse in some nonlinear (nonsuperposable) filtering technique. In this paper the application of a nonlinear filtering technique, viz., Median Filtering introduced by Tukey,[3,4] is discussed. Fig. 1(d) shows the output after filtering input 1(b) by a median filter.

Section II introduces the definition and some properties of medians, Section III discusses the various methods of finding the median and section IV surveys some applications of such a filtering technique. Finally Section V concludes the paper.

II Definition and Properties of Median

Let $\{x_i, 1 \leqslant i \leqslant n\}$, where n is odd be the input samples. The median of $\{x_i, 1 \leqslant i \leqslant n\}$, $M[x(n)]$, is defined as the t th largest x_i where $t = \frac{n+1}{2}$. For example the median of the set $\{x_i\} = \{1,3,1,2,5,7,3,1,1,2,2,\}$ is the 6th largest x_i and is 2. (The linear filter works on the principle of means instead of the medians and in this example the means is $\frac{28}{11}$.)

For an input data $\{x_i, 1 \leqslant i \leqslant \infty\}$ one can define a running median RM by sliding a window of width n, n being odd, and determining the median of the elements in the window. For example consider the data string (1,1,2,1,3, 1,2,2,2,4,5,6,1,3,5,...) than the running medians, with window sizes 3 and 5 are respectively given by RM(3) = {1,1,2,1,2,2,2,2,4,5,5,3,3,...} and RM(5) = {1,1,2,2,2,2,2,4,4,4,5,...}

The median as defined here satisfies the following properties:[5]

(a) Scaling of the input data scales the median by the same factor i.e. $M[\alpha x(n)] = \alpha M[x(n)]$

(b) Median preserves the sharp discontinuities provided the length of the discontinuity is not less than $\frac{n+1}{2}$.

For example consider the data string (0,0,0,0,1,1,1,0,0,0,0,0,1,0,0,0,0,1,1,0,0,0,0,1,1,1,1,1,0,0,0,0,0,). This data string exhibits discontinuities of length 1,2,3 and 5 at samples 13,18,5 and 22, respectively. Running medians of n=3 and n=5 yield, respectively, RM(3) = {0,0,0,1,1,1,0,0,0,0,0,0,0,0,0,1,1, 0,0,0,0,1,1,1,1,1,0,0,0,} and RM(5) = {0,0,0,1,1,1,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,1,1,1,1,1,0,0,0,}. In the former the discontinuity of length 1 is filtered out while in the latter both the discontinuities of lengths 1 and 2 are filtered out.

(c) Outputs of median filters approximately follow input polynomials. Lower order polynomials are more affected by longer medians.

(d) Median smoothers introduce a delay of $(\frac{n-1}{2})$.

Tyan[6] has explored the properties of running median as applicable to monotonic and locally monotonic sequences and has concluded that "A monotonic sequence is invariant under running median and if a sequence is locally monotonic of length m, then it is invariant under a RM(2k+1) where $k \leq m-2$".

Tukey[7] has further suggested a method of determining an approximation to a median. Here the data string is arranged in blocks of 3 and the 3-point medians of the blocks are initially determined. An approximation to the 9-point median, the "Ninther", is obtained by considering the three 3-point medians. For example the data string (1,1,3,3,0,1,2,4,7) yields a 3-point median string of (1,1,4) and a ninther of 1 which is only an approximation to the median of the 9 elements. The exact median of the string is 2. It can be easily seen that if the data string is monotonically increasing or decreasing then the ninther will yield the exact median.

Justusson[8] has investigated the effect of median filtering on impulsive noise superimposed on images. He has shown that if the probability of a cell to be erronous is less than 0.3 then median filtering reduces the number of erronous cells considerably. He also derived the probability of error in the case of salt-and-pepper noise.

The variance of the median of n independent identically distributed points with pdf f(x) and c.d.f. F(x) is (e.g. see Ref. 22)

$$\sigma_\mu^2 = \text{Var (median)} \simeq \frac{1}{4f(\tilde{m})n}$$

where $\tilde{m}$ is the expected median, that is the value of n for which $F(n) = F(\tilde{m}) = 0.5$. If f(x) is normal with mean $\tilde{m}$ and variance σ^2, then variance of the median is

$$\sigma_\mu^2 \simeq \frac{\sigma^2}{n}\frac{\pi}{2}$$

Thus for normal white noise, variance of the median is 57% larger than that of the mean. Hence a moving average or linear filter reduces the variance of normal white noise 57% more than a median filter of the same length. The main advantage of a median filter however, is the preservation of sharp discontinuities and this will be further explored in Section IV.

III Fast Methods of Median Filtering:

Since the introduction of 'Median Filtering' by Tukey, several methods of on-line and off-line median filtering have been proposed. These methods may be classified as:

1. Radix method of Ataman et al[9]
2. Selection network method of Shamos[10], Eversole et al[11] and Narendra[12]
3. Histogram method of Huang et al[13], & Garibotto & Lombselli[14]

The Radix method is based on the binary representation of the elements in the set $\{x_i\}$ and subsequently recognizing the various bits in the word. Let the binary representation of x_i be $(b_1^i b_2^i \ldots b_L^i)$ and that of median $M[x(n)]$ be $(\mu_1 \mu_2 \mu_3 \ldots \mu_L)$. The algorithm for determining the median starts by dividing the elements of the set $\{x_i, 1 \leq i \leq n\}$ into two groups. The first group contains all those elements of the set for which b_1^i is one and the second group the rest of the elements. If a majority of b_1^i, $i = 1, \ldots n$ are equal to 1(0), then $\mu_1 = 1(0)$. This determines the first bit of the median. If $\mu_1 = 1$, then the median M is an element of the set of those elements which have $b_1^i = 1$, say set $S(b_1^i = 1)$. Let the cardinality of this set be C. If all $b_1^i = 1$, i.e. $S(b_1^i = 1) = \{x_i, 1 \leq i \leq n\}$ then obviously M is also the median of the set $S(b_1^i = 1)$. If some $b_1^i = 0$, then M is the t th largest element of $S(b_1^i = 1)$ when $t = \frac{n+1}{2}$. If a majority of $b_1^i = 0$ then the median M is an element of the set $S(b_1^i = 0)$ and is the (t-C)th largest element of this set. Assuming, without loss of generality, that median M is an element of the set $S(b_1^i = 1)$ the algorithm proceeds by operating on set $S(b_1^i=1)$ and subdividing the set based on b_2^i being 1 or 0. The search procedure leads to a tree structure and further details of the method are found elsewhere.[9]

The Selection networks, digital[10,11,15] or analog,[12] are a special case of sorting networks and are arrangements of comparators used for finding the t th largest element of a given set. A comparator module is a device with two inputs and as many outputs. The larger and the smaller of the two inputs appear as the two outputs (Fig. 2(a)). The selection networks for 3-point and 5-point median are shown in Figs. 2(b) and 2(c), respectively.

The method given by Huang et al[10] is based on the histogram of elements in the set $\{x_i, i = 1,2,\ldots n\}$. Let $h_n(a)$ be the number of elements such that $x_i = a$ and let $\lambda_n(w)$ be

$$\lambda_n(w) = \sum_{a=0}^{w} h_n(a)$$

By definition $\lambda_n(M[x(n)]) = \frac{n+1}{2}$. If $\lambda_n(w) = \frac{n+1}{2}$, then w is the median. If $\lambda_n(w)$ is greater/less than $\frac{n+1}{2}$, then w is decremented/incremented by 1 until $\lambda_n(w) = \frac{n+1}{2}$.

The first two methods are suitable for on-line filtering while the last one is essentially for off-line filtering. As can be seen from Fig.2 the minimum number of comparisons for the selection network method for a 3-point and 5-points median are 3 and 7, respectively. However the number of comparisons and the complexity of the selection network increase rapidly with increasing n. The selection network method can be simplified if one is determining the ninther or an approximation to the median of a set. The analog section network suggested by Narendra[12] uses analog selection networks as opposed to digital networks in Shamos method.

Speed and hardware complexity of the radix and the histogram methods does not depend on n but on L, where L is the wordlength of each sample. Worst case running time of the histogram method grows exponentially with L whereas for the radix method it is proportional to L. A detailed comparison of the three methods are discussed elsewhere.[9]

IV Applications of Median Filtering

The two main applications of median filtering are in the fields of speech processing and two-dimensional signal processing such as image processing. As indicated in the introduction median filtering preserves the sharp discontinuities in the signal and hence may be used for applications where the signal in addition to having high frequency contents is corrupted by HF noise. However, median filtering may fail to provide sufficient smoothing of undesirable noise-like components. To overcome this inherent disadvantage Tukey[3] and later Rabiner et al[5] suggest the use of a combination of linear and median smoothers.

Fig. 3 shows a typical arrangement of overall smoothing algorithm. If the input signal $x(n)$ can be written as $x(n) = S[x(n)] + R[x(n)]$ Where S indicates the smooth part and R the rough part of $x(n)$, then, with reference to Fig. 3, the various signals are $u(n) \simeq S[x(n)]$, $v(n) \simeq R[x(n)]$ and $y(n) \simeq S[x(n)] + S[R[x(n)]]$. The delays in Fig. 3 are necessary to compensate for the inherent delays in filtering. Rabiner et al suggest the use of a 3-point median filter and a 3-point Hamming window and have compared the performance of linear smoothers, median smoothers and a combination of median and linear smoothers [Figs. 6-10 of Ref. 5].

In speech processing, measurement and processing errors introduce single or double-point discontinuities and as such median filters are particularly suited for the removal of such errors. Rabiner et al[5] have demonstrated the superiority of a combination of smoothers in the processing of speech intensity data and pitch period contours. In a later communication Jayant[16] has shown yet another application of median filtering. Bit errors in DPCM cause propagating distortion in the decoded waveform. The error is essentially impulsive and the median filter squelches the impulse without smearing the speech waveform. Steele and Goodman[17] have further explored the application of median filtering in smoothing out transmission errors in linear PCM.

The applicability of median filtering to image processing was initially discussed by Pratt[18,19] and Frieden.[20] Since then several authors have considered this application[8,10-13,21]. The ability to detect and recognize edges is a central problem in image processing as edges carry important information. Indeed most images are edge-enhanced before processing. This adds spurious oscillations and noise spikes to the images. A linear post filter for removing the noise smears or blurs the edges. The spurious oscillations in the data can be removed easily by passing the data through a median filter with a window greater than the period of these oscillations Median filtering besides supressing the HF and spiky noise maintains the edge gradient.

In image processing a sliding window covering odd number of picture elements (pixels) is passed across the picture and at every instant the central pixel is replaced by the median of the pixels in the window. Consider an image divided into K^2 picture elements by an K X K grid. The image X is defined by an K X K matrix $X = [x_{ij}]$, $i,j = 1,2,\ldots K$ where x_{ij} are the gray levels of the element in i th row and j th column of the grid. If M is the output of an N X N median filter, then it is defined by an L X L matrix, where $L = K-N+L$, $M = [m_{kl}]$, $\frac{N-1}{2} \leq \{k,l\} \leq K - \frac{N-1}{2}$ and $m_{kl} = \text{Median}\ \{x_{ij}\}$ $i,j = 1,\ldots,N$ and x_{ij} span over the N X N window. Obviously the output of the filter depends on the window shape. Two of the commonly used windows are the square and cross shaped window. (Fig. 4 a and b). Ataman and Alparslan[21] have investigated the effect of those two windows. Fig. 5 a, b,c,d and e show the original 128 x 128 image, original image corrupted by gaussian noise, 3-point median filtered image with square window, 3-point median filtered image with cross window and linear filtered image, respectively.

Narendra[12] has considered processing images by separable filters rather then two-dimensional filters. In the former the pixels in each row of the picture matrix are processed by a one-dimensional window and then the columns are processed by the same window. He shows that the performance, insofar as the smoothing is concerned, of the separable filter is identical to that of a square window filter. He also notes that a separable filter yields a slightly greater variance than its two dimensional counterpart. While a linear smoother performs better for uniform and normal distributions, median filters in general perform better on spiky noise.

He has also compared the retention of corners by the two-dimensional and separable filters [Fig. 7 of reference 12].

V Conclusions

Median filtering offers an alternative approach to processing noisy signals. If the signal contains high frequency components then median filters perform better than linear smoothers in preserving discontinuities in the input. This property of median filters is exploited in their application to image processing. A median filter yields a variance approximately $\frac{\pi}{2}$ times the corresponding linear filter for Gaussin noise and 3 times for uniform noise.

Images contain a large amount of data and as such smoothing of images in realtime requires processing of enormous amount of data in a very short period of time. Median filters do not involve any arithmetic operation and hence have a potential for high-speed application.

In conclusion median filters offer an efficient signal processing algorithm for both one and two dimensional signals and many aspects of median filtering are yet unexplored.

REFERENCES

[1] A.V. Oppenheim and R.W. Schafer, Digital Signal Processing, Prentice-Hall, 1975.

[2] L.R. Rabiner and B. Gold, Theory and Application of Digital Signal Processing. Prentice-Hall, 1975.

[3] J.W. Tukey, "Nonlinear (nonsuperposable) methods for smoothing Data," in Cong. Rec., 1974 EASCON, p. 673.

[4] A.E. Beaton and J.W. Tukey, "The Fitting of power series, meaning polynomials, illustrated on band-spectroscopic data," Technometrics, vol. 16, pp. 147-185, May 1974.

[5] L.R. Rabiner, M.R. Sambur, and C.E. Schmidt, "Applications of a nonlinear smoothing algorithm to speech processing," IEEE Trans. Acoust Speech, Signal Processing, vol. ASSP-23.

[6] S.G. Tyan, "Fixed points of running medians," (unpublished report), Dept. of Electrical Engineering and Electrophysics, Polytechnic Institute of New York, Brooklyn, N.Y., 1977.

[7] J.W. Tukey, "The ninther, a technique for low-effect robust (resistant) location in large samples" in H.A. David, ed. Contributions to Survey Sampling and Applied Statistics, Academic Press, New York, pp. 251-257, 1978.

[8] B. Justusson, "Noise reduction by median filtering," Department of Math, Royal Inst. of Tech. Report No: TRITA-MAT-1978-7, Stockholm, Sweden, Feb. 1978.

[9] E. Ataman, V.K. Aatre, M.K. Wong, "A fast method for real-time median filtering", 1979 (to appear).

[10] M.I. Shamos, "Robust picture processing operators and their implementation as circuits," Proc. of Image Understanding Workshop, Pittsburg Penn., pp. 127-129, Nov. 1978.

[11] W.L. Eversole, D.J. Mayer, F.B. Frazee, and T.F. Cheek, Jr., "Investigation of VLSI technologies for image processing," Proc. of Image Understanding Workshop, Pittsburg, Penn., pp. 191-195, Nov., 1978.

[12] P.M. Narendra, "A separable median filter for image noise smoothing", Proc. of 1978 Conf. on Pattern Recognition and Image Processing, Chicago, Ill., pp. 137-141, May 1978.

[13] T.S. Huang, G.T. Yang and G.Y. Tang, "A fast two-dimensional median filtering algorithm," IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP-27, pp. 13-18, Feb. 1979.

[14] G. Garibotto and L. Lambrelli, "Fast on-line implementation of two-dimensional median filtering," Electronic Letters, Vol. 15, pp. 24-25, January, 1972.

[15] D.L. Knuth, Sorting and Searching, Addison-Wesley, Reading, Mass., 1975.

[16] N.S. Jayant, "Average and Median based smoothing techniques for improving digital speech quality in the presence of transmission errors," IEEE Trans. Commun., Vol. COM-24, pp. 1043-1045, Sept. 1976.

[17] R. Steele and D.J. Goodman, "Detection and Selective smoothing of transmission errors in linear PCM," Bell System Tech. J., Vol. 56, pp. 300-409, March 1977.

[18] W.K. Pratt, "Median filtering" in Semiannual report, USC IPI report 620, Image Processing Institute, University of Southern California, Los Angeles, Calif., pp. 116-120, 1975.

[19] W.K. Pratt, Digital Image Processing, Wiley-Interscience, New York, N.Y. 1978.

[20] B.R. Frieden, "A new restoring algorithm for the preferential enhancement of edge gradients," J. Opt. Soc. Amer., Vol. 66, pp.280-283, 976.

[21] E. Ataman and E. Alparslan, Application of Median Filtering Algorithm to Images, Technical Report No. UI-78/10, Electronics Research Division, Marmara Research Institute, Turkey.

[22] H.A. David, Order Statistics, John Wiley & Sons, 1970.

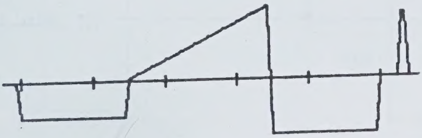


Fig. 1 (a)

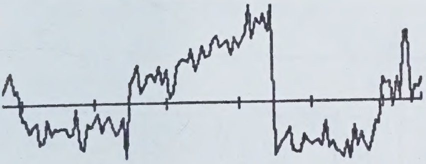


Fig. 1 (b)

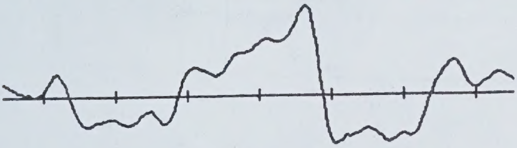


Fig. 1 (c)

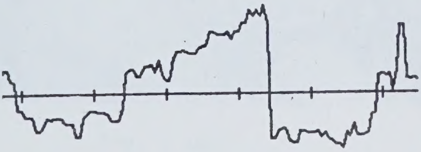


Fig. 1 (d)

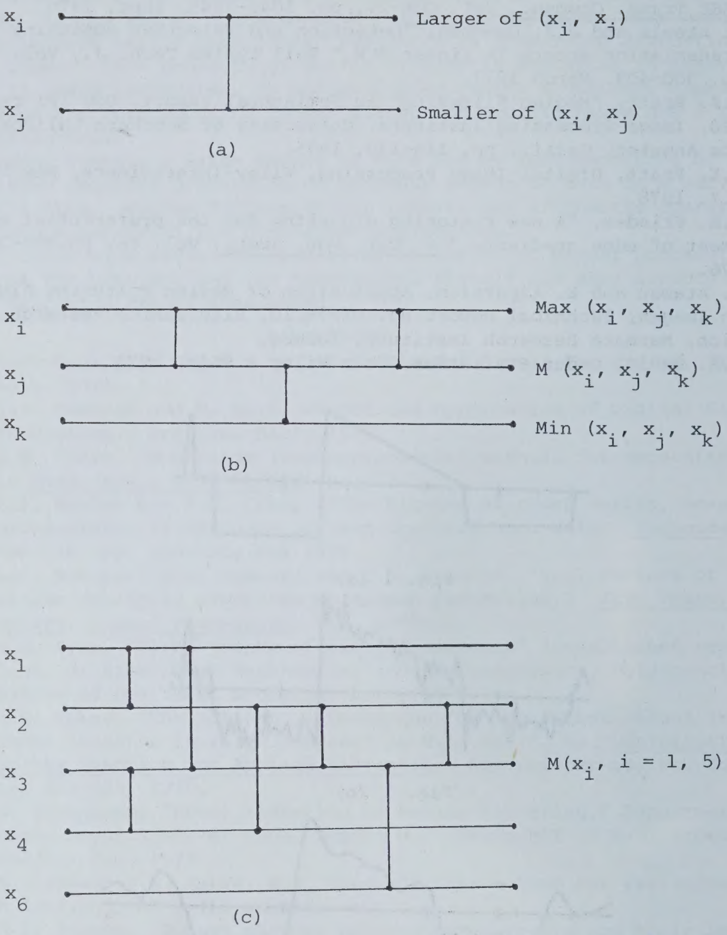

x_i
x_j
Larger of (x_i, x_j)
Smaller of (x_i, x_j)
(a)
x_i
x_j
x_k
Max (x_i, x_j, x_k)
M (x_i, x_j, x_k)
Min (x_i, x_j, x_k)
(b)
x_1
x_2
x_3
x_4
x_6
$M(x_i, i = 1, 5)$
(c)

Fig. 2

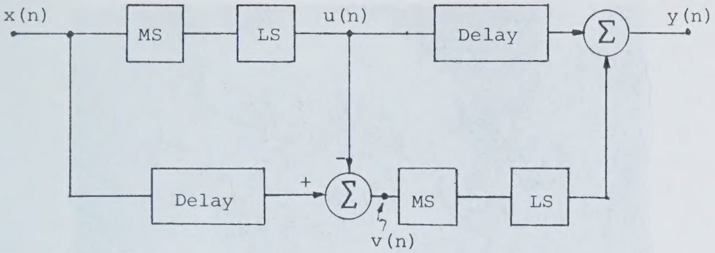
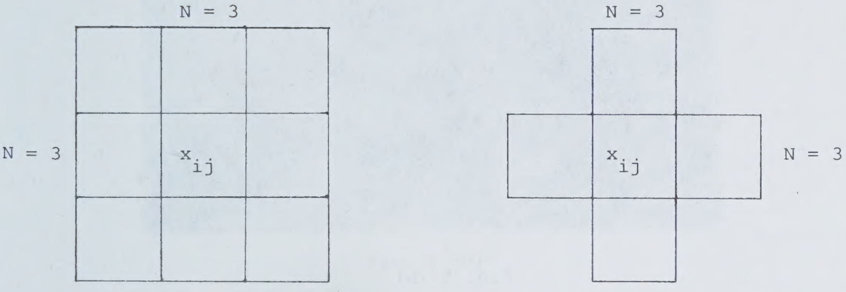

x(n)
MS
LS
u(n)
Delay
Σ
y(n)
−
+
Delay
Σ
MS
LS
v(n)

FIG. 3


N = 3
N = 3
x_{ij}
N = 3
N = 3
x_{ij}

(a)

(b)

Fig. 4

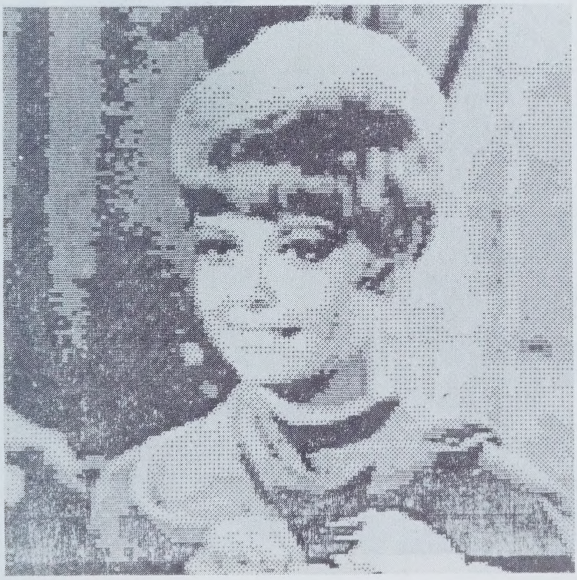


Fig. 5 (a)

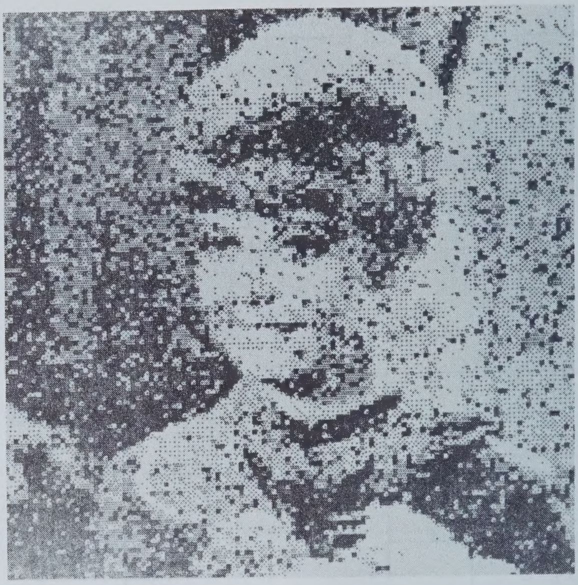


Fig. 5 (b)

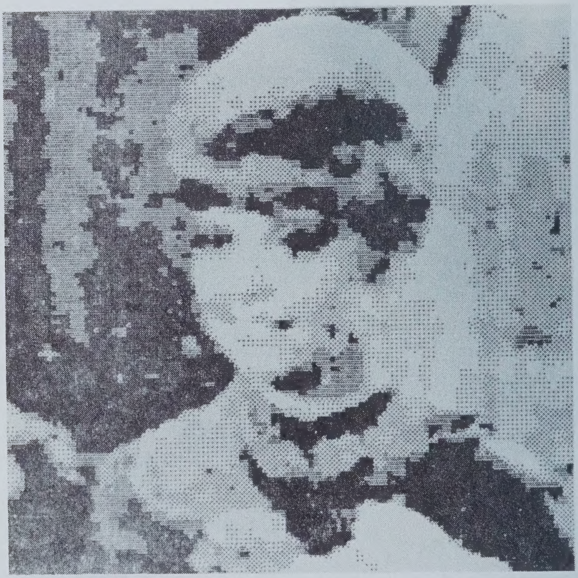


Fig. 5 (c)

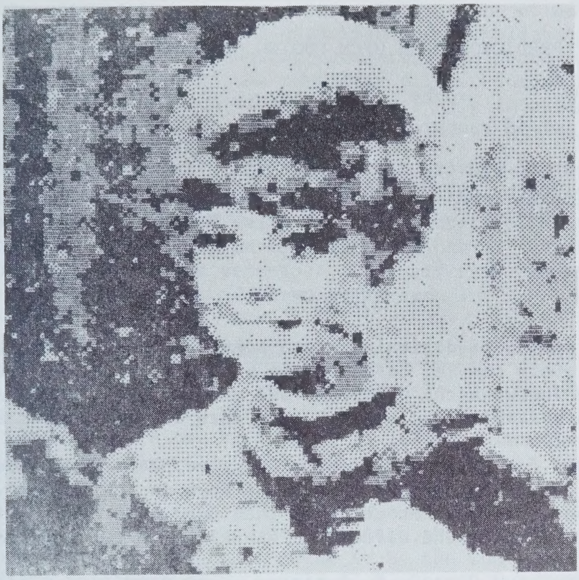


Fig. 5 (d)

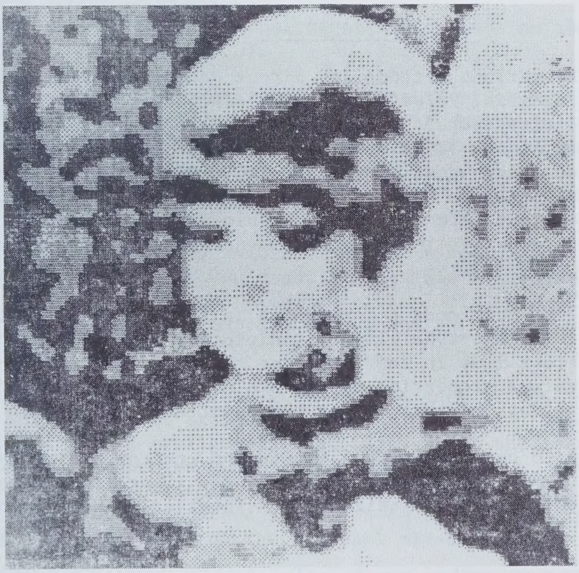


Fig. 5 (e)

EDGE-ADAPTIVE PREDICTORS FOR DPCM CODING OF PICTURES

by

P. COHEN
Department of Electrical Engineering
Ecole Polytechnique de Montréal
Montréal, Québec, Canada

A. VALETTE
Institut de Recherche de
l'Hydro-Québec
Varennes, Québec, Canada

ABSTRACT

In DPCM coding of digital picture signals, a prediction is made for the value of every sample, based on the past history of the encoded signal. The difference between the predicted and actual values is then quantized into one of n possible levels and a code word is transmitted accordingly.

This paper describes two locally adaptive predictors whose originality consists of detecting the presence of an edge in the vicinity of every picture element. Such a detection is then used in order to minimize the instantaneous value of the difference signal to be transmitted.

Using this kind of adaptive predictors in DPCM coders, instead of invariant type predictors, leads to a smaller number of required quantizing levels, while preserving the quality of the coded pictures.

INTRODUCTION

Among the various methods for direct coding of digitized picture signals, Differential Pulse Code Modulation (DPCM) methods are characterized by simplicity and efficiency. Their operating principle is illustrated in Figure 1. A prediction for the value of every sample is made, based on the past history of the encoded signal, not the original picture signal itself. The difference between the predicted and actual values is quantized into one of n possible levels and a code word is transmitted in order to indicate to the decoder which level effectively occurred.

Most of the DPCM coders mentionned in the litterature [1,2] make use of invariant-type linear predictors. A main limitation of these predictors lies in the fact that they are optimized only with respect to average properties of picture signals, which make them blind to local variations of these signals properties.

In order to overcome such a limitation, the predictor can be made adaptive to the local context of the signal. Some predictors of this type have been already investigated and interesting performance where obtained.[3]

This paper describes two locally adaptive predictors whose originality consists of observing first the context (i.e. the neighboring elements) of every picture element in order to detect the presence of an edge in the vicinity of this element. Once this detection is accomplished, the predictor chooses, among the neighboring elements, which ones are to be considered in the computation of the predicted value. Such a procedure tends to improve the performance of the predictor by eliminating all the neighbors whose luminance is not close to the luminance of the current element.

The first predictor to be described is a one-dimensional adaptive predictor in which the predicted value for each picture element is equal to the luminance of one of its four immediate neighbors. The second predictor is based on a similar principle, but this time the predicted value is made of a weighted sum of three neighbors.

ONE-DIMENSIONAL EDGE-ADAPTIVE PREDICTOR

Since a digitized picture can be adequately represented in each of its directions by a first-order Markov process, the context to be considered for an element X can be reduced to its four immediate neighbors A, B, C, D according to Figure 2. The context of element A itself consists of the elements E, F, B, C.

The predictor being considered here is such that, at each sampling instant, the prediction value, P, is going to be the luminance of one of the four immediate neighbors of X, namely A, B, C or D. Accordingly, the difference signal to be transmitted is given by:

$$e = X - P \text{ with } P = A \text{ or } B \text{ or } C \text{ or } D \tag{1}$$

where the choice between these four alternatives depends on the particular configuration of (X, A, B, C, D).

Since it is desirable to minimize the amplitute of the difference signal, an ideal solution would consist of choosing as prediction element the immediate neighbor whose luminance is the nearest to X. In this case, the prediction value P would be such that:

$$|X - P| = \min \left[|X - A|, |X - B|, |X - C|, |X - D| \right] \tag{2}$$

However, although it is always possible to make the coding loop operate according to expression (2), it would not be possible to restore uniquely the value of X from the difference signal, unless additional information is given, indicating from which immediate neighbor the difference e was taken.

Nevertheless, a rather accurate guess can be made of the immediate neighbor of X which verifies expression (2), by performing some sort of extrapolation based on the configuration prevailing, at the preceding sampling instant, for A and its own neighbors E, F, B and C. Such a prediction can be performed identically in the coding loop and the decoding loop, thus giving a decoded signal which is an exact replica (ignoring the quantizing noise) of the original signal.

Thus, starting with element X and its neighbors A, B, C, D, the prediction method consists of progressively eliminating some of these neighbors by using adequate selection tests. This elimination goes until only one neighbor remains; this element is then called the "Estimated Nearest Neighbor of X" (ENN) and its luminance intensity is taken as the prediction value P.

The procedure used in searching for the ENN, can be described as follows:

Step 1: Seperation of A, B, C, D into two sets G_1 and G_2, by means of adequate separation tests. This is done in order to classify the context elements into two amplitude categories. The only cases considered are those for which G_1 and G_2 contain adjacent elements namely:

Case 1: $G_1 = (A)$; $G_2 = (B, C, D)$
Case 2: $G_1 = (A, B)$; $G_2 = (C, D)$
Case 3: $G_1 = (A, B, C)$; $G_2 = (D)$

Discriminating between those three cases necessitates two comparison tests.

Step 2: Determination of the set (either G_1 or G_2) which is more likely to contain the nearest neighbor of X. This determination is performed by means of a predictive test .

Step 3: Elimination of the set which did not verify the preceding test.

- If the remaining set contains only one element, this element is the ENN.
- If the remaining set contains more than one element, these elements are separated into two new sets G_1 and G_2, by means of a separation test, and return is made to Step 2.

The nature of the decision tests involved in this procedure results from an optimisation of the predictor performance over a class of representative television pictures. This optimisation consists of making the predictor find as often as possible the effective nearest neighbor of X.

Figure 3 shows the predictor flow-chart thus determined. In this flow-chart the notation UV represents the absolute value of the amplitude difference between elements U and V.

Predictor performance

The predictor previously described has been compared with two well-known linear predictors and with the "ideal predictor" which would operate according to expression (2).

Table 1 illustrates some results obtained from these comparisons. It can be noticed that the proposed predictor has a definite superiority over the two linear predictors.

TWO-DIMENSIONNAL EDGE - ADAPTIVE PREDICTOR

The predictor which has been described above can be viewed as a generalization of a fixed one-dimensional predictor, in which the prediction value is always taken equal to the luminance of the previous element A.

In a similar way, the predictor to be described now is a generalization of a fixed two-dimensional predictor [2,4] in which the prediction value is always computed by the expression

$$P = A + C - B \tag{3}$$

Referring to Figure 2, the adaptive predictor chooses optionally as prediction value P, one of the three values:

$$\begin{aligned} P_1 &= A + B - F \\ P_2 &= A + C - B \\ P_3 &= A + D - C \end{aligned} \tag{4}$$

The choice between these three values depends upon the presence of an edge in the vicinity of element X and upon its direction.

Here again, an ideal solution would consist of choosing, as prediction value, the value P_1, P_2 or P_3 which minimizes the amplitude of the difference signal:

$$|X-P| = \mathrm{Min}\,[\,|X-P_1|\,,\ |X-P_2|\,,\ |X-P_3|\,] \tag{5}$$

Unfortunately, this procedure would not lead, in general, to a uniquely decodable difference signal.

Different strategies have been tried in order to preserve decodability while transmitting a difference signal as close as possible to the minimum of equation (5). The most successful strategies which have been studied operate according to the following principle:

(1) in the coding loop, the difference signal e_T to be transmitted is always generated according to equation (5). Although such a signal is not uniquely decodable in general, it has been observed that unique decodability does happen in particular situations (approximately 18% of all cases, as measured on typical television pictures).

(2) in the decoding loop, whenever e_T is uniquely decodable, the value of element X can be exactly restituted. Whenever e_T is not uniquely decodable, the receiving end predictor estimates which one of P_1 or P_2 or P_3 was most probably used, at the transmitting end. This estimation is based on an observation of the current context, namely (A, B, C, D, E, F) and on the decision (P_1, P_2 or P_3) corresponding to the previous sampling instant.

Whenever this estimation is correct, the value of element X is exactly restituted. However, if an error occurs in the estimation, the decoded value of X may eventually differ from the original value. In this last case, the difference between the original and decoded values must be taken into account in the coding loop, in order to avoid systematic error propagation.

This operation principle is illustrated by the flow-chart of Figure 4.

Several estimation procedures have been tried for the situations where e_T is not uniquely decodable. Table 2 illustrates comparative results for the most successful procedure. In this table, Predictor No.1 is a fixed two-dimensional predictor (where P is always equal to A+C-B) and predictor No.2 is the ideal predictor corresponding to equation (5). Here again, a definite superiority of the proposed predictor can be observed. over the fixed two-dimensional predictor.

CONCLUSION

This paper has presented two edge-adaptive predictors which can be successfully used in DPCM loops for black-and-white picture signals. These predictors show better performance, in term of the difference signal statistics, than invariant-type linear predictors.

Better performance is also obtained in terms of slope-overload noise, because of the local edge detection involved in the adaptation procedure. Consequently, the number of required quantizing levels may be reduced, while preserving the quality of the decoded pictures.

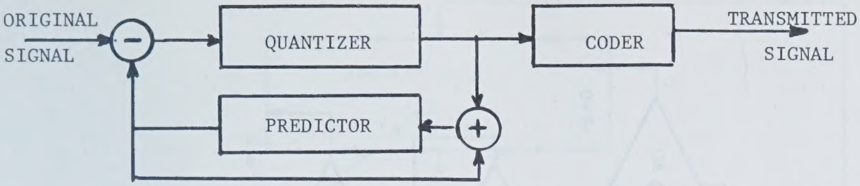
REFERENCES

[1] Connor, D.J., Brainard, R.C., Limb, J.O.: "Intraframe Coding for Picture Transmission" IEEE Proceedings, Special Issue on Digital Picture Processing, Vol. 60, No.7, (July 1972).

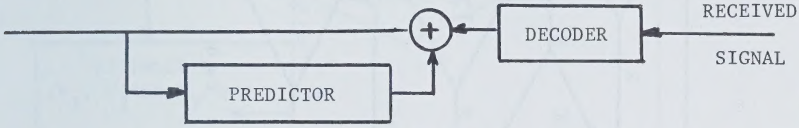
[2] O'Neal, J.B.: "Predictive Quantizing System (Differential Pulse Code Modulation) for the Transmission of Television Signals", Bell System Technical Journal, Vol. 45, (May-June 1966).

[3] Connor, D.J., Pease, R.F.W., Scholes, W.G.: "Television Coding Using Two-Dimensional Spatial Prediction", Bell System Technical Journal, Vol.50, no. 3, (March 1971).

[4] Habibi, A., Robinson, G.S.: "A Survey of Digital Picture Coding", Journal of the IEEE Computer Society, Vol.7, no.5, (May 1974).


ORIGINAL
SIGNAL
QUANTIZER
CODER
TRANSMITTED
SIGNAL
PREDICTOR

(a) CODING LOOP


DECODER
RECEIVED
SIGNAL
PREDICTOR

(b) DECODING LOOP

FIGURE 1 - DPCM CODING SYSTEM

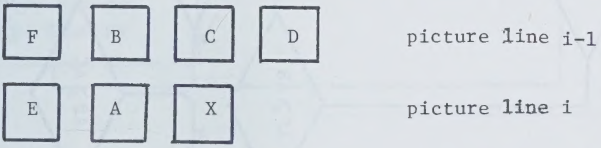

F
B
C
D
picture line i-1
E
A
X
picture line i

Figure 2 - REPRESENTATION OF THE CONTEXT OF ELEMENT X

	Predictor 1 e=X-A	Predictor 2 e=X-A-C+B	Predictor 3 (equ. 2)	Proposed Predictor
Picture 1 (Group)	58.7	25.8	1	23.6
Picture 2 (cameraman)	8.0	8.2	1	4.5
Picture 3 (Portrait)	15.5	11.0	1	4.0

TABLE 1 - ONE-DIMENSIONAL PREDICTOR: MEAN SQUARE VALUE OF THE DIFFERENCE SIGNAL.

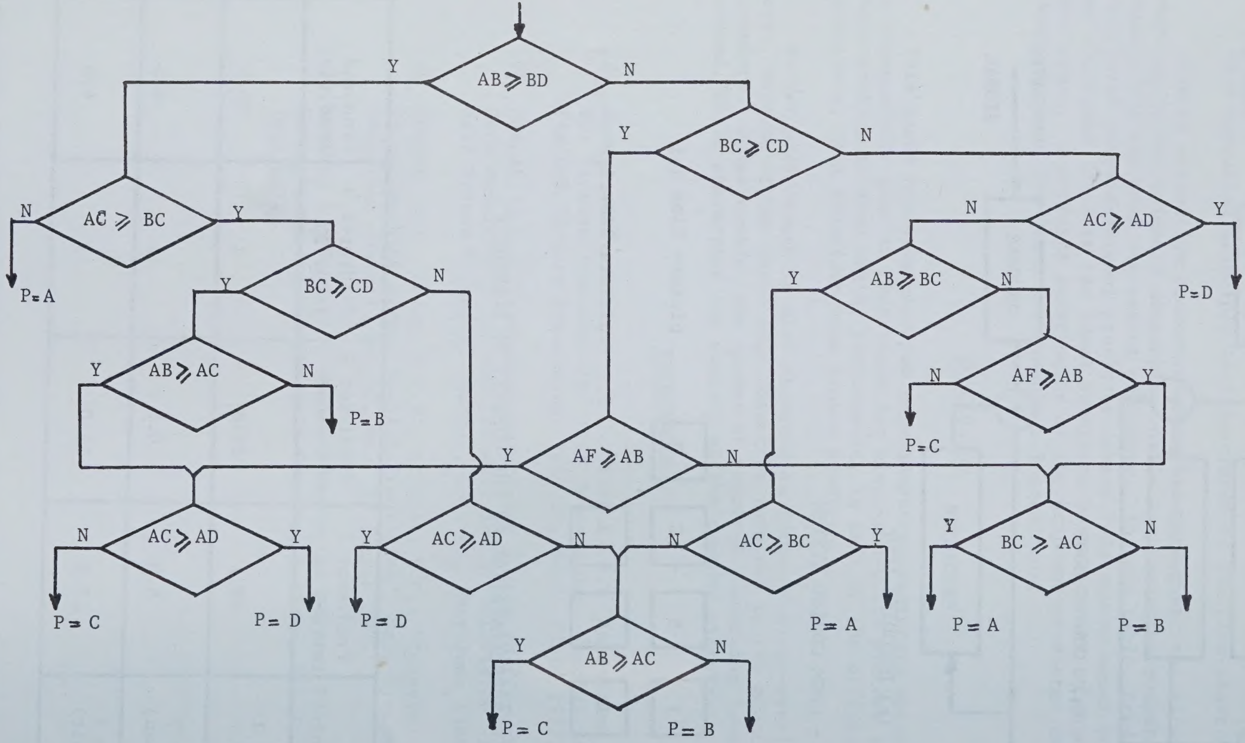

AB ⩾ BD
Y
N
BC ⩾ CD
Y
N
AC ⩾ BC
N
Y
P=A
AC ⩾ AD
N
Y
P=D
BC ⩾ CD
Y
N
AB ⩾ BC
Y
N
AB ⩾ AC
Y
N
P=B
AF ⩾ AB
N
Y
P=C
Y
AF ⩾ AB
N
AC ⩾ AD
N
Y
P= C
P= D
Y
AC ⩾ AD
N
P= D
N
AC ⩾ BC
Y
P= A
Y
BC ⩾ AC
N
P= A
P= B
Y
AB ⩾ AC
N
P= C
P= B

FIGURE 3 - ONE DIMENSIONAL PREDICTOR FLOW-CHART.

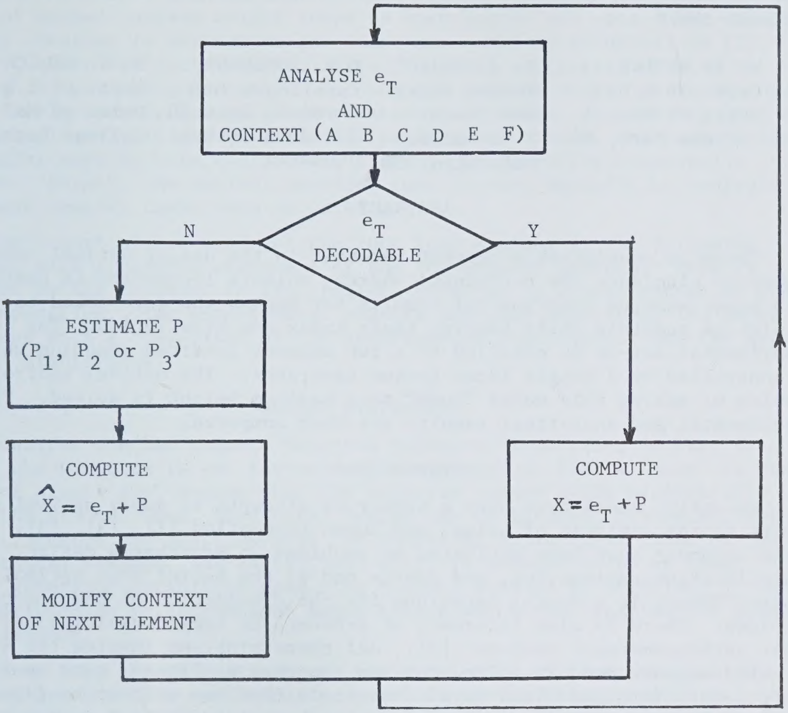

ANALYSE e_T AND CONTEXT (A B C D E F)
e_T DECODABLE
N
Y
ESTIMATE P (P_1, P_2 or P_3)
COMPUTE $\hat{X} = e_T + P$
MODIFY CONTEXT OF NEXT ELEMENT
COMPUTE $X = e_T + P$

FIGURE 4 - TWO-DIMENSIONAL PREDICTOR FLOW-CHART.

	Predictor 1 $e_T = X-A-C+B$	Predictor 2 (equ.5)	Proposed Predictor
Picture 1 (Group)	1.74	1.0	1.10
Picture 2 (Cameraman)	2.01	1.0	1.33
Picture 3 (Portrait)	2.08	1.0	1.20

TABLE 2 - TWO-DIMENSIONAL PREDICTOR: MEAN-SQUARE VALUE OF THE DIFFERENCE SIGNAL.

SOME EXPERIMENTAL, ANALYTICAL AND COMPUTATIONAL RESULTS CONCERNING MAXIMAL VERTICAL JUMPS

W. S. LEVINE
Dept. of E.E.
Univ. of Md.
College Park, Md.

F. E. ZAJAC*
Rehab. Engrg.
R&D Center
V. A. Hosp.
Palo Alto, CA

M.R. ZOMLEFER
Karolinska Inst.
Physiol. Dept. III
114 33 Stockholm
Sweden

M. R. BELZER
Dept. of E.E.
Univ. of Md.
College Park, Md.

ABSTRACT

There is considerable current interest in the use of optimal control theory to elucidate the techniques whereby animate locomotion is controlled. This paper reports experimental results for humans who were asked to jump as high as possible while keeping their knees and hips locked. The experimental set-up is modelled by a two segment inverted pendulum which is controlled by a single ideal torque generator. The optimal control problem of making this model "jump" to a maximum height is solved. The experimental and analytical results are then compared.

INTRODUCTION

Recently, there have been a number of attempts to apply optimal control theory to the analysis of animal and human locomotion [1], [2], [3]. These attempts have been motivated by problems in prosthesis design, rehabilitation engineering, and sports and by the belief that optimal control theory is a useful technique for the elucidation of complex control problems. There is also interest, at present, in legged vehicles [4] and other anthropomorphic devices [5]. All these problems involve the dynamics of multi-segment pendula. The problems are thus nonlinear, have unusual state constraints and often involve controls that are neither bang-bang nor linear. Because of this, it has generally been impossible to solve these optimization problems analytically. The exception to this has been a recent paper [6] in which the simplest case, that of making a baton "jump" as high as possible, was solved analytically.

The baton problem is too simple to be a reasonable model of any physiological control problem. The next simplest theoretical problem is to make a two segment inverted pendulum, controlled by a single ideal torque generator, "jump" as high as possible. It is possible to experimentally approximate this theoretical problem by asking humans to jump as high as possible while keeping their knee and hip joints locked. This paper compares the results of just such an experiment with the results of an analysis of the two segment, ideal torque controlled, maximum height "jumper".

The results of this study of the two segment or "foot jump" give useful information about the maximum torque which is generated by the triceps surae (calf) muscle group in humans and about physiological bang-bang controls. However, the major reason for studying the foot jump was to

---

Acknowledgement:

This research has been supported in part by N.I.H. Grant #NS11971, in part by Swedish Medical Research Council Projects No. 3026 and 5402 to Professor Sten Grillner, and in part by the Computer Science Center of the University of Maryland. The experimental research was carried out at Karolinska Institute, in Professor S. Grillner's Laboratory, where Dr. Zajac was a visiting scientist from August '78 to June '79.

*Formerly Associate Professor, Department of E. E., University of Maryland.

develop and sharpen the tools needed for the experimental and theoretical study of normal maximum height jumps by both humans and cats. The reasons for our interest in maximum height jumps are outlined in detail in [3]. Briefly, there are four reasons. First, jumping is closely related to running (or galloping in cats). Second, the maximum height vertical jump has a fairly unambiguous performance measure, thereby, simplifying both the theoretical analysis and experimental procedures. Third, the task is done in similar ways by cats and humans, thereby, facilitating comparative studies. Fourth, the maximal vertical jump is more amenable to analysis than more complex tasks such as walking.

The report of our study of the foot jump begins, in the following section, with a careful statement of the theoretical problem. This is followed by the solution to the problem, obtained by a combination of theory and computation. The experimental results are then described. The paper concludes with a comparison of the experimental observations and the analytical predictions.

PROBLEM STATEMENT

Consider the two segment inverted pendulum illustrated in Fig. 1. Assume the pendulum is not fastened to the ground so that a "jump" is possible. For simplicity, assume that the pendulum cannot slide horizontally along the ground. Thus, horizontal forces can be ignored. Otherwise, the masses, lengths and inertias will be chosen to approximate those of the experimental subject. The controller, u(t), is assumed to be an ideal torque generator capable of producing torques bounded by $0 \le u(t) \le u_{max}$, where u_{max} corresponds to the maximum ankle torque which the experimental subject could produce.

Given this physical structure, the problem is to find the u(t) $(0 \le t \le T)$ which will cause this mechanism to "jump" to the maximum possible height. This must be accomplished without violating either of the constraints (1) neither segment can go through the ground and (2) $-\theta \le \phi \le \pi + \theta$.

It is not trivial to translate the above physical problem into a meaningful mathematical optimization problem. The mathematical problem requires four unambiguous components: (a) dynamics, (b) initial and terminal conditions, (c) constraints and (d) performance criterion. It will be seen that there is considerable interaction among (a), (b) and (c).

In particular, consider the dynamics. There are two distinct cases that are important. The first case is illustrated in Fig. 1 and corresponds to only the toe on the ground. The second case corresponds to the whole foot on the ground. One might think it is also necessary to consider the situation where the entire device is airborne. It will be assumed that the objective is to maximize the peak height achieved by the center of mass of the object. This height is completely determined at the instant of lift-off. Hence, there is no need to consider the dynamics once the object is airborne.

Case 1: Only the "toe" on the ground. The dynamics are given by

$$[I_1 + m_1\ell_1^2 + m_2(\ell_1 + \ell_2)^2]\ddot{\theta} + m_2(\ell_1 + \ell_2)\ell_3 \cos(\theta + \phi)\ddot{\phi}$$

$$- m_2(\ell_1 + \ell_2)\ell_3 \sin(\theta + \phi)\dot{\phi}^2$$

$$+ (m_1\ell_1 + m_2(\ell_1 + \ell_2))\, g \cos\theta = u(t) \qquad (1)$$

$$\begin{aligned}\left[I_2 + m_2\ell_3^2\right]\ddot{\phi} + m_2(\ell_1 + \ell_2)\ell_3 \cos(\theta + \phi)\ddot{\theta}& \\ - m_2(\ell_1 + \ell_2)\ell_3 \sin(\theta + \phi)\dot{\theta}^2& \\ + m_2\ell_3\, g \cos\phi = u(t)& \qquad (2)\end{aligned}$$

It is convenient to include a third equation here. This is the equation for F_v, the force exerted by the toe on the ground.

$$F_v = (m_1 + m_2)g + m_1\ddot{y}_1 + m_2\ddot{y}_2 \qquad (3)$$

where

$y_1(t)$ = vertical position of c.m. of foot

$y_2(t)$ = vertical position of c.m. of trunk

thus

$$\begin{aligned}F_v = (m_1 + m_2)g + m_1\ell_1(-\sin\theta\, \dot{\theta}^2 + \cos\theta\, \ddot{\theta})& \\ + m_2(\ell_1 + \ell_2)(-\sin\theta\, \dot{\theta}^2 + \cos\theta\, \ddot{\theta})& \\ + m_2\ell_3(-\sin\phi\, \dot{\phi}^2 + \cos\phi\, \ddot{\phi})& \qquad (4)\end{aligned}$$

Case 2: Foot on the ground. The dynamics simplify to

$$\ddot{\phi}(t) = [u(t) - m_2 g\ell_3 \cos\phi(t)] \;/\; [I_2 + m_2\ell_3^2] \qquad (5)$$

$$\theta(t) = \dot{\theta}(t) = \ddot{\theta}(t) = 0 \qquad (6)$$

$$F_v = (m_1 + m_2)g + m_2\ell_3(\cos\phi\, \ddot{\phi} - \sin\phi\, \dot{\phi}^2) \qquad (7)$$

It is important to know the boundary between case 1 and case 2. This is easily expressed in terms of the center of pressure ℓ_{cp} (see [7] for more details)

$$\ell_{cp} = \ell_1 + \ell_2 - \frac{(m_1 g\ell_2 + u)}{F_v} \qquad (8)$$

The "foot" stays flat on the ground while $\ell_{cp} \geq 0$.

The "heel" leaves the ground when $\ell_{cp} < 0$.

Notice that the controller, at certain states, can choose between the dynamics represented by (1) and (2) and the dynamics represented by (5) and (6). This is caused by the non-holonomic constraint that the heel can rest on the ground.

Next, consider (b), the initial and terminal conditions. The choice of initial condition is quite straightforward. Experimentally, the subjects begin by standing with their feet flat on the ground and their trunk approximately vertical. In the model, this is an equilibrium point ($\theta = 0$, $\phi = 90^\circ$). More to the point, all stationary initial conditions (states for which $\dot{\theta} = \dot{\phi} = 0$) can be reached from this one (provided we allow ϕ to be perturbed by $\pm\, \epsilon$).

The terminal time is the instant that the mechanism leaves the ground. Since the toe, or foot, rests on the ground, this corresponds to the first instant at which $F_v < 0$. Obviously, one does not want an open set for the target set. However, in the Baton problem [6] there are optimal trajectories along which $F_v = 0$. Furthermore, F_v depends on the control u(t) through the dependence of $\ddot{\theta}$ and $\ddot{\phi}$ on u. Thus, for some states, the terminal time depends explicitly on the value of the control at that instant. Fortunately, there is a way out of the dilemma. The next section begins with a proof that the terminal condition is

$$F_v\,(\theta,\ \dot{\theta},\ \phi,\ \dot{\phi},\ u_{max}) = 0 \tag{9}$$

Now consider (c) constraints. The first constraint is

$$0 \le u(t) \le u_{max} \qquad ,\ \text{all } t \tag{10}$$

u(·) piecewise continuous

A second constraint is imposed by the need to keep the toe on the ground until Eq. (9) is satisfied. Thus, there is a constraint

$$F_v(\theta(t),\phi(t),\dot{\theta}(t),\dot{\phi}(t),u(t)) \ge 0 \tag{11}$$

Note that the physical constraint imposed by the ground does not appear among the mathematical constraints. Instead, it appears as a complicated nonlinearity in the dynamics.

Finally, the performance criterion, item (d) is to maximize

$$J(t_f,u) = y_c(t_f) + \frac{1}{2g}\,\dot{y}_c^2(t_f) \tag{12}$$

where t_f = the instant of lift-off

$y_c(t)$ = the vertical position of the center of mass

or,

$$J(t_f,u) = \frac{1}{(m_1+m_2)} \Big\{ (m_1\ell_1 + m_2(\ell_1 + \ell_2)) \sin\theta + m_2\ell_3 \sin\phi + \frac{1}{2g(m_1+m_2)} \big[(m_1\ell_1 + m_2(\ell_1 + \ell_2)) \cos\theta\ \dot{\theta} + m_2\ell_3 \cos\phi\ \dot{\phi}^2 \big] \Big\} \tag{13}$$

This completes the formulation of the optimal control problem. To summarize, we want to maximize (13), evaluated at the instant at which (9) is satisfied, subject to the constraints (10) and (11) and the dynamics. Note that the dynamics are given by (1) and (2), or (5) and (6).

SOLUTION

The problem is solved in three parts. First, we prove that (9) is the terminal condition and, concomitantly, that $u(t_f)=u_{max}$. Second, we show that the control $u(t)=u_{max}$ is optimal "near" lift-off. This result is then extrapolated backwards in time to the instant of heel lift-off. Finally, we determine the set of states that are reachable from the initial state $(\theta=\dot{\theta}=\dot{\phi}=o,\phi=90^{o})$ while keeping the foot flat on the ground. The optimal control is then to go from initial state to heel lift-off in whatever manner is convenient. Then, from heel lift-off to toe lift-off the optimal control is $u(t)=u_{max}$ and the optimal trajectory is the unique corresponding trajectory.

First, the performance criterion, in the form given by Eq. (12) is unchanged when multiplied by the constant $(m_1+m_2)g$. Differentiating with respect to t_f gives

$$\frac{dJ(t_f)}{dt_f} = mg\dot{y}_c(t_f)+m\ddot{y}_c(t_f)\dot{y}_c(t_f)$$
$$=\dot{y}_c(t_f)[mg+m\ddot{y}_c(t_f)]$$
$$=\dot{y}_c(t_f)F_v(t_f)$$

where (3) is used to simplify the expression. It is obvious that $\dot{y}_c(t_f)>0$ for an optimal jump and that $F_v \geq 0$ for the "toe" to be on the ground. Thus, $J(t_f)$ has an extremum when $\frac{dJ(t_f)}{dt_f} = 0$ and is monotone increasing prior to t_f. Thus, in any state for which it is possible to make $F_v(\theta,\dot{\theta},\varphi,\dot{\varphi},u)>0$ it is advantageous to do so. Therefore, all optimal jumps occur on the curve in the state space corresponding to

$$\max_{0\leq u\leq u_{max}} F_v(\theta,\dot{\theta},\varphi,\dot{\varphi},u)=0 \tag{14}$$

Substituting (1) and (2) into (4) shows that F_v is linear in u. It is possible to show that the coefficient of u, which depends on θ and φ, is positive in most of the state space. It is easy to see that a sufficient condition for this coefficient to be positive is that $0\leq\theta,\varphi\leq\pi/2$ and $(\theta+\varphi)\geq\pi/2$. This region includes all of the possible states at which an optimal jump might leave the ground. Thus, (9) is in fact the terminal condition for an optimal jump and $u(t_f)=u_{max}$.

The next step in deriving the optimal control is to extend the above result backwards in time from the instant of lift-off. To do this it is convenient to formulate a new optimal control problem that is valid only in a subset of the state space. In particular, assume that once the jumper's heel leaves the ground it is not advantageous to return the heel to the ground before lift-off. While this is certainly not true for normal jumps it does appear to be true for the constrained jumps analyzed here. Futhermore, in the case of the normal jump there is experimental evidence that energy storage mechanisms, not included in our model, are the source of the advantage.

Having made the above assumption, we now temporarily restrict our attention to states in which the heel is off the ground ($\theta>0$). For these states the dynamics are given by (1) and (2). The performance criterion is still given by (13), the terminal condition by (9) and the control constraint by (10). The mixed constraint given by (11) still applies and greatly complicates the application of the necessary conditions of the minimum principle. Furthermore, we are fairly sure that (11) is not an active constraint just prior to lift-off. Thus, we apply the minimum principle to the "relaxed" problem obtained by ignoring the constraint (11). As long as the optimal control for this "relaxed" problem does not violate the constraint (11) we know it is the optimal control for the true problem as well.

The necessary conditions then become, besides (1) and (2) in state space form, that

$$H(p_1{}^*(t), p_2{}^*(t), p_3{}^*(t), p_4{}^*(t), \theta^*(t), \phi^*(t), \dot{\theta}^*(t), \dot{\phi}^*(t), u^*(t)) = 0 \tag{15}$$

$$\dot{\underline{p}}^*(t) = \frac{-\partial H}{\partial \underline{x}}\Big|_* \tag{16}$$

with the condition that

$$u^*(t) \text{ maximizes } H(\underline{p}^*, \underline{x}^*, u) \tag{17}$$
$$0\leq u\leq u_{max}$$

and the boundary condition

$$\underline{p}^*(t_f) = \frac{\partial J}{\partial \underline{x}}\Big|_* + \alpha \frac{\partial Fv}{\partial \underline{x}} (\underline{x}^*(t_f), u_{max}) \tag{18}$$

when

$$\underline{x} = [\theta\ \dot{\theta}\ \varphi\ \dot{\varphi}\ ]^T$$

and α is a constant, to be determined.

We have not been able to find an explicit solution for the necessary conditions. However, there is a straightforward technique for using the computer to obtain a solution. The technique is as follows:

(1) Pick any point $\underline{x}$ satisfying (9). Such an $\underline{x}$ is the terminal state of an optimal jump.

(2) Determine $\underline{p}^*(t_f)$, to within a constant α, from (18).

(3) Determine α from (15).

(4) Integrate (1) and (2) and (16) backwards, with $u(t)=u_{max}$ while checking (17) to ensure that $u^*(t)=u_{max}$

(5) The first state at which either u_{max} does not satisfy (17) or $\theta=0$ is on the boundary of the region where $u^*(t)=u_{max}$.

The above calculation can be repeated until the boundary of the region where $u^*(t)=u_{max}$ is established. We have not yet done this. However, we have done the above calculation from our candidate optimal terminal state to verify its optimality. In that case, $u(t)=u_{max}$ all the way back to $\theta=0$ (heel on the ground). It is conjectured, but is not yet proven, that the same result ($u(t)=u_{max}$ once heel-off occurs) is valid for all jumps that are close to optimal.

The last step in deriving the optimal control is to compute the set of states that can be reached from the initial state ($\theta=\dot{\theta}=\phi=0,\phi=90^\circ$) while keeping the heel on the ground. This set is shown in Fig. 2. The derivation of Fig. 2 is based on (5)-(8), the equations describing the dynamics when the heel is on the ground. The crucial requirement is that ℓ_{cp} (see (8)) remain greater than or equal to zero so the heel stays on the ground. As long as the heel is on the ground the dynamics are given by (5) and, as shown in [6], the phase plane trajectories can be calculated analytically.

In deriving Fig. 2 it is convenient to start with the effect of the constraint

$$\ell_{cp}(\theta,\dot{\theta},\varphi,\dot{\varphi},u) = \ell_{cp}(0,0,\varphi,\dot{\varphi},u) \geq 0 \tag{19}$$

In region I, at the bottom of the figure, it is impossible for (19) to be satisfied. Basically, the centrifugal force is large enough to pull the heel off the ground independent of the control u. The boundary of region I is given by

$$\ell_{cp}(0,0,\varphi,\dot{\varphi},0) = 0 \tag{20}$$

The second effect of the constraint (19) is illustrated by the roughly elliptical curve in Fig. 2. In the region exterior to this curve, the heel will leave the ground provided the control u(t) satisfies

$$u(t) > u_a$$

The curve itself satisfies

$$\ell_{cp}(0,0,\varphi,\dot{\varphi},u_a) = 0$$

Similar curves can be drawn for all admissible values of u. Values of u larger than u_a make the ellipse smaller (finally collapsing to the point C and then vanishing entirely. Smaller values of u expand the ellipse until it breaks apart and takes a shape like the boundary of region I (with a symmetric about $\dot{\phi}=0$ part that is not shown). The value of u_a is significant for two reasons:

(1) u_a is the largest torque that can be exerted in the initial state without raising the heel off the ground.

(2) u_a is also the torque required to maintain the equilibrium position ($\dot{\theta}=\dot{\phi}=0$) in which $\theta=0$ and ϕ is the minimum possible. This equilibrium point is A in Fig. 2 and, it should be noted, it is impossible to maintain equilibrium for smaller values of ϕ.

Once the role of the constraint is understood, it is fairly straightforward to calculate the set of reachable states. The lower boundary, curve OF, is just the trajectory that results from falling forward, under the influence of gravity, with u=0. We have already asserted that points on the $\dot{\phi}=0$ axis to the left of A are unreachable. It is easy to show that A is reachable. It is also easy to show that the upper boundary of the set of states that are reachable with the heel on the ground corresponds to the trajectories which start at A and apply the maximum torque possible without causing the heel to lift-off. This maximum possible torque is calculated from (19) with the current state substituted for ϕ and $\dot{\phi}$. Thus, one obtains a feedback control in which u(t) is neither u_{max} nor 0. Thus, AB is the trajectory that results from starting just to the left of A and falling at the minimum possible rate. Trajectory AD is obtained by starting just to the right of A and accelerating at the maximum possible rate consistent with (19). All states in the interior of the region bounded by OFBADO are reachable.

Having determined the set of reachable states with the heel on the ground, all that remains is to piece together the optimal control and trajectory. To do this, we assumed that $u=u_{max}$ is always optimal from heel lift-off on. We then calculated the jump height that would be achieved by starting in a grid of reachable states with the heel on the ground and applying $u=u_{max}$ from there until lift-off. The maximum height was obtained by starting at A. We then proved, numerically, that $u(t)=u_{max}$ satisfies the necessary conditions with initial condition A. Thus, the optimal control is to maneuver, by any one of many possible controls to state A. From state A until lift-off the optimal control is $u(t)=u_{max}$.

EXPERIMENTAL RESULTS

The basic experiment was to ask a human subject to jump, as high as he could, while keeping all his joints, with the exception of the ankle joint, locked. For safety's sake none of the joints were physically restrained so there was some movement in most of the joints. The experiment was repeated in two subjects with essentially identical results. The subject jumped from a Kistler force plate so accurate values for the vertical force F_v, horizontal force F_h and ℓ_{cp} were obtained at each instant of time. The trajectory of the subject was recorded and digitized via the Selspot system. From this data, θ and ϕ were calculated. $\dot{\theta}$ and $\dot{\phi}$ were obtained by numerical differentiation (after low-pass filtering). The torque, u(t), was then computed from the equations of motion. Of course, these calculations depend on knowing the body parameters as accurately as possible and so these were measured.

The experimental and theoretical results for one subject are shown in Figs. 3 and 4. All calculations are based on the following parameter values: $m_1=2.2$ kg., $m_2=73.87$ kg., $\ell_1=.095$ m., $\ell_2=.08$ m., $\ell_3=1.11$ m.,

I_1=.008 kg. m^2, I_2=12.072 kg. m^2. The height of the optimal jump was theoretically 1.38 m. and experimentally 1.27 m. The theoretical u_{max} was 230.9.

One large difference between the experiment and the theory is easily correctable. The experimental ankle angle is defined by the knee, ankle and toes so this angle is approximately 115° when the subject is standing upright. In the theoretical model this same situation has 90° for the "ankle angle".

There are several other large differences that are more fundamental. In particular, there is a large difference between the experimental and theoretical angular velocities. This leads to a large difference in maximum jump height. The theoretical jumper jumps .27 m. higher than his initial height. The experimental jumper jumps .11 m. above his initial height. All of these differences can be traced to the differences in the torques shown in Fig. 3. The experimental subject simply cannot exert torque equal to u_{max} over the full range of ankle angles. At large ankle angles the human cannot exert large ankle torques. Similarly, the human cannot exert bang-bang torque. There is a rather long delay in the build up of muscle force. Thus, we believe that the experimental subject is exerting maximum possible torque from about t=900 msec. until lift-off (the first data point to lift-off).

Three final comments are in order relating experimental and theoretical results. First, the experimental subject appears to cheat slightly as evidenced by the increase in F_v prior to the exertion of ankle torque. Second, the switch times and duration of the take-off show fairly good agreement between experiment and theory. Third, and finally, we have a great deal of additional experimental data which will be reported elsewhere.

CONCLUSIONS

There are several conclusions to be drawn from these results. First, it is clear that the theoretical model should be improved by the inclusion of a more accurate model of torque generation. We are presently replacing the ideal torque generator by an ideal force generator connected as is the triceps surae muscle group (the calf). This should cause u_{max} to decrease as the ankle angle approaches 180°. If necessary, we will also add an angle dependent retarding torque to model the effect of elastic elements in the joint. Finally, if necessary, we will add a better model of force generation to account for the build up of torque.

Second, the division of the problem into two parts, one with heel on and one with heel off greatly facilitates the solution. The dynamics are too complex to expect an analytic solution for the optimal control. However, the program described earlier offers a numerical verification of the necessary conditions in the heel off region.

Third, the extension of the heel off analysis to more realistic jumps appears straightforward. A reasonable model for such a jump requires at least a four segment inverted pendulum. For such a model the heel on analysis is much more complicated. We are presently working on this analysis.

REFERENCES

1. Chow, D. and D. Jacobson, "Studies of Human Locomotion Via Optimal Proramming", Mathematical Biosciences, Vol. 10, 1971, pp. 239-306.

2. Hatze, H., "The Complete Optimization of a Human Motion", Mathematical Biosciences, 28, 1976, pp. 99-135.

3. Zajac, F.E. and W.S. Levine, "Novel Experimental and Theoretical Approaches to Study the Neural Control of Locomotion and Jumping", in Posture and Movement: Perspective for Integrating Sensory and Motor Research on the Mammalian Nervous System, edited by R.E. Talbot and

D.R. Humphrey, Raven Press, New York, 1979.

4. McGhee, R.B., "Robot Locomotion", Proceedings of the Int'l. Conf. on Neural Control of Locomotion, Phila., Pa. 1975.

5. Stepanenko, Yu, and Vukobratovic, M., "Dynamics of Articulated Open-Chain Active Mechanisms", Mathematical Biosciences, 28, 1976, pp. 137-170.

6. Roberts, W.M., Levine, W.S., and Zajac, F.E., "A Model for Maximum Height Jumping", (to appear) IEEE Trans. on AC, Vol. AC-24, No. 5, October, 1979.

7. Cho, Y.M., "Optimal Control of Two Segment Limb Model", M.S. Thesis, Univ. of Maryland, Dept. of E.E., 1978.

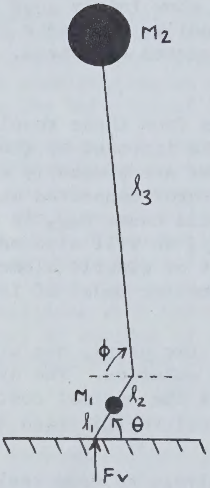


Fig. 1 - Mathematical model and notation

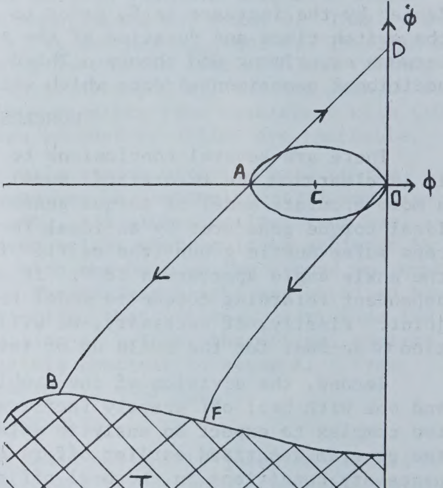


Fig. 2 - State space and reachable states with heel on the ground

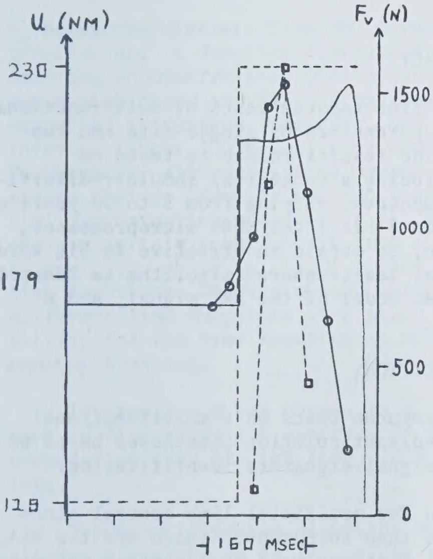


Fig. 3 - Torque and vertical force versus time

— F_v	theoretical
-o- F_v	experimental
--- U^*	theoretical
-▫- U^*	experimental

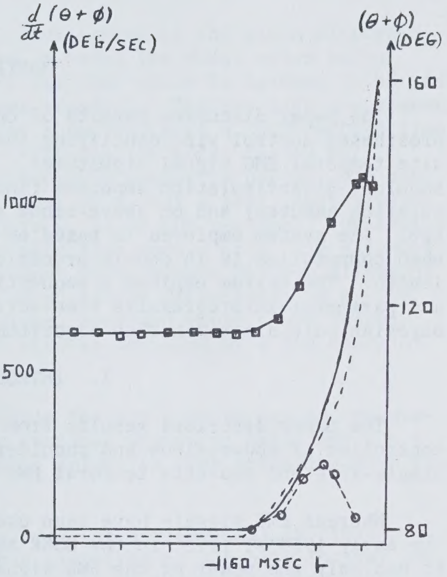


Fig. 4 - Ankle angle and its derivative versus time

— $\theta+\phi$	theoretical
-▫- $\theta+\phi$	experimental
--- $\dot\theta+\dot\phi$	theoretical
-o- $\dot\theta+\dot\phi$	experimental

CONTROL OF MULTIFUNCTIONAL ABOVE-ELBOW AND SHOULDER-DISARTICULATION PROSTHESES BY SINGLE-SITE EMG TEMPORAL SIGNATURE IDENTIFICATION

D. GRAUPE, J. SALAHI AND J. GROSSPIETSCH
Department of Electrical Engineering
Illinois Institute of Technology
Chicago, IL 60616

ABSTRACT

The paper discusses results of on-line amputee tests of multifunctional prostheses control via identifying the parameters of single-site and two-site temporal EMG signal signatures. The results relate to tests on shoulder-disarticulation amputees (including a congenital shoulder-disarticulation amputee) and on above-elbow amputees, varying from 5 to 50 years of age. The system employed is based on an 8-bit Intel 8080 microprocessor, when computation is in double precision, to obtain an effective 16-bit word length. The system employs a sequential least-squares algorithm to identify a 4 parameter autoregressive time-series model of the EMG signal, and a Bayesian rule discrimination algorithm.

I. INTRODUCTION

The paper describes results from amputee tests on a multifunctional controller of above-elbow and shoulder-disarticulation prostheses based on single-site and two-site temporal EMG signal signature identification.

Whereas EMG signals have been used for artificial limb control since the early 1950's, first in the USSR and then in Europe, Canada and the USA, it was only the power of the EMG signal that was used as a control parameter rather than the complete temporal (time-related) pattern of that signal. Since the power in the signal is a single signal-parameter, this has resulted in only few multifunctional control applications,(i.e. applications to controlling several limb functions at several joints, as is required in cases of above-elbow and shoulder-disarticulation amputations). In these multifunctional cases, a many-electrode approach (about 10 electrode sites) was used, as in the multifunctional versions of the Boston-arm, i.e. the Utah multifunctional arm and the Swedish multifunctional arm [1], [2], [3], where the spatial distribution of the signal's power was considered for function discrimination, i.e. for deciding which limb function is to be activated. Only in the advent of fast and relatively powerful microcomputers was it possible to consider the temporal EMG pattern thus replacing the information as obtained from the signal's power at many sites with the many parameters of the stochastic temporal pattern of the signal at one (or two) sites.

Possibly the stochastic nature of that pattern (in contrast to other biomedical applications of pattern recognition, as in computerized EKG analysis, where the signal considered is basically deterministic), was also responsible to the fact that temporal EMG signals were not used in prosthesis control. The work on using that temporal signature for multifunctional prosthesis control as presently reported was started in the present first author's laboratory in 1973 [2], and has by now lead to a microprocessor-based prototype laboratory-system that has been tested on eight different amputees, always successfully after only a few hours of training. It is this system that will be further discussed below.

II. THE MULTIFUNCTIONAL CONTROLLER

The control system considered is based on identifying the parameters of a pure AR (autoregressive) time-series model of the EMG signal, given by:

$$y_k = a_1 y_{k-1} + \ldots a_n y_{k-n} + w_k, \quad k=0,1,2\ldots \tag{1}$$

y_k being the discrete time EMG signal as measured at the given electrode location and k denoting discrete time, n being the model order and w_k denoting an uncorrelated (white noise) residual which is assumed to be wide sense stationary and SME (second moment ergodic). The two last requirements cannot be guaranteed. However, one can show that over a short enough time interval (about 100 m.sec) wide sense stationarity holds fairly well. Furthermore, although second moment ergodicity cannot be proven, one can assume that we identify not the true model of y_k but a best least-squares (LS) approximation model as in (1) where w_k is an SME residual. Otherwise one can certainly not obtain from a single time record of y_k the parameters a_i ($i=1,2,\ldots n$) as in (1), i.e. one cannot discriminate between different limb functions, discrimination being based on the differences in a_i for different limb functions at a given signal site and their close repeatability for the same function at that site as achieved by a few hours of amputee-training.

In practice n was chosen as 4, since for $n > 4$ there was no improvement in discrimination accuracy, or in residual whiteness while for $n > 4$ computation time was too long, such that delays in limb actuation were too long.

The identification algorithm employed was a sequential LS (least squares algorithm as in [4], since that algorithm is not only (p-lim) convergent under the conditions of (1), but, due to the near maximum-likelihood (ML) properties of (1) it is close to being statistically efficient, having the fastest convergence-rate possible [3]. Furthermore, sequential LS algorithms are also least sensitive to round-off errors arising in finite word-length computers such as must be used in (light-weight) patient-borne microprocessor realizations. We comment that many other LS identifiers have the convergence-rate properties of sequential LS but are sensitive to finite word-length errors. Indeed, the square-root algorithms are as robust to finite word-length as is the sequential LS algorithm. However, the sequential LS algorithm is considerably faster (in computation time) per iteration than square root algorithm [5]. Now, since for adequate real-time operation, limb actuation delays of > 0.2 seconds are not acceptable, and since an 8 bit microprocessor was used (in double precision) the sequential LS identifier was the obvious choice.

The data sequence that is employed for identification is of 80-100 m.sec duration, to effectively cover the spectrum over which information exists in the temporal EMG pattern, i.e., from 100 Hz to 1200 Hz. Hence, an ADC (analog-digital convertor) sampling rate of 2500 per second was used. The above has yielded adequate convergence of the identification algorithm with 150-200 samples of y_k per decision (identification). We note that longer sequences would lead to too long limb actuation delays. Also, even though some information may lie beyond the spectrum range considered, as desired. Since the gradient search techniques can only find a local extremum, a two-step parameter estimation procedure has been suggested [20, 22]. As a first step, one obtains a "coarse" parameter estimate on a finite parameter set which is an appropriate approximation to the (original) continuous (infinite) uncertain parameter set; then, in the second stage, gradient search techniques may be used to "refine" the parameter estimate in some neighborhood of the coarse estimate. The true parameter may not belong to the finite parameter set after quantization of the continuous uncertain parameter set. Another situation of interest arises when one wishes to approximate a higher order system with a lower order model due to e.g., limitations on the available computation and storage capabilities. We use maximum likelihood as a criterion for choosing a model from Θ which best fits the model represented by θ_0, given input-output measurements $u(t)$ and $y(t;\omega)$, $0 \leq t \leq T$.

Theorem 1 defines the limit set of the ML estimate of θ on Θ as $T \to \infty$. The limit set Θ_ℓ may be redefined as follows

$$\Theta_\ell = \left\{\theta : \arg\left(\min_{\theta\in\Theta} q(\theta) - q(\theta_0)\right)\right\} .$$

Therefore, the ML identification technique leads to the selection of a model $\theta \in \Theta$ which asymptotically minimizes the difference $q(\theta) - q(\theta_0)$.

Remark 3: It should be pointed out that (particularly in case of the model approximation problem) various θ's in Θ may correspond to matrices A, B, C, D, F, and G which are of different dimensions (possibly lower) than the "true" matrices corresponding to θ_0. It is a tacit assumption in this paper that conditions (C1) - (C3) hold in this case too with obvious modifications; e.g., in (C2), rank of the observability matrix is "full" (which may not be n).

V. CONCLUDING REMARKS

The problem of global convergence of the maximum likelihood estimates of unknown parameters of a continuous-time stochastic linear system has been investigated when the observation noise covariance is unknown. The unknown parameter set was assumed to be finite. Convergence was proved under certain sufficient condition called the Identifiability Condition. As in [3], we considered convergence in probability; however, we considered absolute maximum of the likelihood functional, whereas in [3], consistency of a "local" maximum in a sufficiently small neighborhood of the true parameter value was proved. Moreover, we also investigated the case when the true model is not included in the unknown parameter set.

REFERENCES

[1] A. V. Balakrishnan, *Stochastic Differential Systems I*. Berlin: Spring-Verlag, 1973.

[2] A. V. Balakrishnan, *J. Opt. Theory Appl.*, *9*, pp. 187-212, 1972.

[3] A. Bagchi, *Automatica*, *11*, pp. 533-536, Sept. 1975.

[4] A. Bagchi, Ph.D. Thesis in Engr., Univ. of Calif., Los Angeles, 1975.

[5] A. Bagchi, IMA Conf., Oxford, England, Sept. 1978.

[6] K. W. Iliff and R. E. Maine, *J. of Guidance & Control*, *2*, pp. 228-234, May-June, 1979.

[7] E. Tse and J. J. Anton, *IEEE Trans. Automatic Control*, AC-17, pp. 637-646, Oct. 1972.

[8] K. J. Åström and P. Eykoff, *Automatica*, *7*, pp. 123-162, 1971.

[9] R. K. Mehra and D. G. Lainiotis (eds.), *System Identification: Advances and Case Studies*. New York: Academic Press, 1976.

[10] G. C. Goodwin and R. L. Payne, *Dynamic System Identification: Experiment Design and Data Analysis*. New York: Academic Press, 1977.

[11] Special Issue, *IEEE Trans. Automatic Control*, AC-19, Dec. 1974.

[12] R. B. Ash, *Real Analysis and Probability*. New York: Academic Press, 1972.

[13] R. Bellman, *Introduction to Matrix Analysis*, New York: McGraw-Hill, 1960.

[14] J. L. Doob, *Stochastic Processes*, New York: Wiley, 1953.

[15] E. Wong, *Stochastic Processes in Information and Dynamical Systems*, New York: McGraw-Hill, 1971.

[16] S. Kullback, *Information Theory and Statistics*, New York: Wiley, 1959.

[17] H. Cramer, *Mathematical Methods of Statistics*, Princeton, N.J.: Princeton Univ. Press, 1946.

[18] A. Wald, *Annals Math. Statistics*, Vol. 20, pp. 595-601, 1949.

[19] J. Wolfowitz, *Annals Math. Statistics*, Vol. 20, pp. 601-602, 1949.

[20] Y. Baram and N. R. Sandell, Jr., *IEEE Trans. Automatic Control*, AC-23, pp. 61-66, Feb. 1978.

[21] Y. Baram, *IEEE Trans. Automatic Control*, AC-23, pp. 958-960, Oct. 1978.

[22] R. M. Hawkes and J. B. Moore, *Proc. IEEE*, *64*, pp. 1143-1150, Aug. 1976.

[23] B. D. O. Anderson, J. B. Moore, and R. M. Hawkes, *Automatica*, *14*, pp. 615-622, Nov. 1978.

[24] L. Ljung, in *System Identification: Advances and Case Studies*, R. K. Mehra and D. G. Lainiotis, ed., New York: Academic, 1976.

[25] P. E. Caines, *IEEE Trans. Automatic Control*, AC-21, pp. 500-505, Aug. 1976.

have negligible actuation delays, and fast 16-bit microprocessor such as an Intel 8086 microprocessor.

REFERENCES

[1] R. W. Mann: Keynote Address, Session 8, Eng. Foundation Conf. on Biomechanics of Movement, Henniker, NH, June 1979.

[2] Graupe, D. and Cline, W.: IEEE Trans. on Sys. Sci. & Cybernetics, pp. 252-258, March 1975.

[3] Graupe, D., Magnussen, J. and Beex, A.: IEEE Trans. on Aut. Cont., pp. 538-544. Aug. 1978.

[4] Graupe, D.: Identification of Systems, 2nd Edition, R. Krieger Publishing Co., Huntington, NY, 1976.

[5] Graupe, D, Jain, V. K. and Salahi, J.: Report # EE-Sys-2-78, Elect. Eng. Dept., IIT, Chicago, 1978.

MODELING HUMAN PERFORMANCE IN AN ANTI-AIRCRAFT TRACKING TASK WITH DELAYED MEASUREMENTS

KUANG-CHUNG WEI
Systems Research Laboratories, Inc.
Dayton, Ohio 45440

MARIS VIKMANIS
A.F. Aerospace Medical Research Lab., WPAFB, Ohio 45433

Luenberger observer theory has been successfully applied to model human tracking performance in an anti-aircraft artillery system, versus an optimal control approach. In the event of a full manual control mode, the gunner plays both the role of a tracker and a lead angle computer. The compensatory tracking task is greatly complicated by the inclusion of lead angle estimation. In order to supplement the task, an additional measurement channel is made available to the gunner, in this experiment, through the miss distance of tracers from the target trajectory. The tracer dynamics enter the man-machine system in a delayed fashion and causes a great deal of concerns in the development of a human model with a simple structure.

In this paper, we focus on the theoretical derivation of the underlying dynamical equations and the design of a gunner model consisting of an observer and a controller. Since the closed-loop equations contain delayed dynamics, the exact solution for the delay differential equations is not available. An algorithm based on the average approximation method is derived to approximate the delay differential equations by a set of ordinary differential equations. The order of the ordinary differential equations is determined by the order of the original system and the order of approximation. The resulting differential equations are then discretized and integrated on a CDC 6600 digital computer.

There are a total of eight parameters, two observer gains, three feedback controller gains and three coefficients of the remnant covariance function to be identified in the model. A direct search algorithm, based on Rosenbrock's hill-climbing optimization method, is devised to identify these parameters systematically while minimizing the least-squares error between the model output and the empirical data.

The basic formulation proposed in this paper may also be applied to model the human's tracking performance for other operational tasks, e.g., tracking with interrupted measurements. The gunner model has the merit of structural simplicity, and therefore less requirements for computer storage as well as running time. The parameter identification scheme takes advantage of a direct search method and identifies model parameters regardless of the order of the approximated system.

The Radix method is based on the binary representation of the elements in the set $\{x_i\}$ and subsequently recognizing the various bits in the word. Let the binary representation of x_i be $(b_1^i b_2^i \ldots b_L^i)$ and that of median $M[x(n)]$ be $(\mu_1 \mu_2 \mu_3 \ldots \mu_L)$. The algorithm for determining the median starts by dividing the elements of the set $\{x_i, 1 \leq i \leq n\}$ into two groups. The first group contains all those elements of the set for which b_1^i is one and the second group the rest of the elements. If a majority of b_1^i, $i = 1, \ldots n$ are equal to 1(0), then $\mu_1 = 1(0)$. This determines the first bit of the median. If $\mu_1 = 1$, then the median M is an element of the set of those elements which have $b_1^i = 1$, say set $S(b_1^i = 1)$. Let the cardinality of this set be C. If all $b_1^i = 1$, i.e. $S(b_1^i = 1) = \{x_i, 1 \leq i \leq n\}$ then obviously M is also the median of the set $S(b_1^i = 1)$. If some $b_1^i = 0$, then M is the t th largest element of $S(b_1^i = 1)$ when $t = \frac{n+1}{2}$. If a majority of $b_1^i = 0$ then the median M is an element of the set $S(b_1^i = 0)$ and is the (t-C)th largest element of this set. Assuming, without loss of generality, that median M is an element of the set $S(b_1^i = 1)$ the algorithm proceeds by operating on set $S(b_1^i=1)$ and subdividing the set based on b_2^i being 1 or 0. The search procedure leads to a tree structure and further details of the method are found elsewhere.[9]

The Selection networks, digital[10,11,15] or analog,[12] are a special case of sorting networks and are arrangements of comparators used for finding the t th largest element of a given set. A comparator module is a device with two inputs and as many outputs. The larger and the smaller of the two inputs appear as the two outputs (Fig. 2(a)). The selection networks for 3-point and 5-point median are shown in Figs. 2(b) and 2(c), respectively.

The method given by Huang et al[10] is based on the histogram of elements in the set $\{x_i, i = 1,2,\ldots n\}$. Let $h_n(a)$ be the number of elements such that $x_i = a$ and let $\lambda_n(w)$ be

$$\lambda_n(w) = \sum_{a=0}^{w} h_n(a)$$

By definition $\lambda_n(M[x(n)]) = \frac{n+1}{2}$. If $\lambda_n(w) = \frac{n+1}{2}$, then w is the median. If $\lambda_n(w)$ is greater/less than $\frac{n+1}{2}$, then w is decremented/incremented by 1 until $\lambda_n(w) = \frac{n+1}{2}$.

The first two methods are suitable for on-line filtering while the last one is essentially for off-line filtering. As can be seen from Fig.2 the minimum number of comparisons for the selection network method for a 3-point and 5-points median are 3 and 7, respectively. However the number of comparisons and the complexity of the selection network increase rapidly with increasing n. The selection network method can be simplified if one is determining the ninther or an approximation to the median of a set. The analog section network suggested by Narendra[12] uses analog selection networks as opposed to digital networks in Shamos method.

Speed and hardware complexity of the radix and the histogram methods does not depend on n but on L, where L is the wordlength of each sample. Worst case running time of the histogram method grows exponentially with L whereas for the radix method it is proportional to L. A detailed comparison of the three methods are discussed elsewhere.[9]

IV Applications of Median Filtering

The two main applications of median filtering are in the fields of speech processing and two-dimensional signal processing such as image processing. As indicated in the introduction median filtering preserves the sharp discontinuities in the signal and hence may be used for applications where the signal in addition to having high frequency contents is corrupted by HF noise. However, median filtering may fail to provide sufficient smoothing of undesirable noise-like components. To overcome this inherent disadvantage Tukey[3] and later Rabiner et al[5] suggest the use of a combination of linear and median smoothers.

Fig. 3 shows a typical arrangement of overall smoothing algorithm. If the input signal $x(n)$ can be written as $x(n) = S[x(n)] + R[x(n)]$ Where S indicates the smooth part and R the rough part of $x(n)$, then, with reference to Fig. 3, the various signals are $u(n) \simeq S[x(n)]$, $v(n) \simeq R[x(n)]$ and $y(n) \simeq S[x(n)] + S[R[x(n)]]$. The delays in Fig. 3 are necessary to compensate for the inherent delays in filtering. Rabiner et al suggest the use of a 3-point median filter and a 3-point Hamming window and have compared the performance of linear smoothers, median smoothers and a combination of median and linear smoothers [Figs. 6-10 of Ref. 5].

In speech processing, measurement and processing errors introduce single or double-point discontinuities and as such median filters are particularly suited for the removal of such errors. Rabiner et al[5] have demonstrated the superiority of a combination of smoothers in the processing of speech intensity data and pitch period contours. In a later communication Jayant[16] has shown yet another application of median filtering. Bit errors in DPCM cause propagating distortion in the decoded waveform. The error is essentially impulsive and the median filter squelches the impulse without smearing the speech waveform. Steele and Goodman[17] have further explored the application of median filtering in smoothing out transmission errors in linear PCM.

The applicability of median filtering to image processing was initially discussed by Pratt[18,19] and Frieden[20]. Since then several authors have considered this application[8,10-13,21]. The ability to detect and recognize edges is a central problem in image processing as edges carry important information. Indeed most images are edge-enhanced before processing. This adds spurious oscillations and noise spikes to the images. A linear post filter for removing the noise smears or blurs the edges. The spurious oscillations in the data can be removed easily by passing the data through a median filter with a window greater than the period of these oscillations Median filtering besides supressing the HF and spiky noise maintains the edge gradient.

In image processing a sliding window covering odd number of picture elements (pixels) is passed across the picture and at every instant the central pixel is replaced by the median of the pixels in the window. Consider an image divided into K^2 picture elements by an $K \times K$ grid. The image X is defined by an $K \times K$ matrix $X = [x_{ij}]$, $i,j = 1,2,\ldots K$ where x_{ij} are the gray levels of the element in i th row and j th column of the grid. If M is the output of an $N \times N$ median filter, then it is defined by an $L \times L$ matrix, where $L = K-N+L$, $M = [m_{kl}]$, $\frac{N-1}{2} \leq \{k,l\} \leq K - \frac{N-1}{2}$ and $m_{kl} = \text{Median}\,\{x_{ij}\}$ $i,j = 1,\ldots,N$ and x_{ij} span over the $N \times N$ window. Obviously the output of the filter depends on the window shape. Two of the commonly used windows are the square and cross shaped window. (Fig. 4 a and b). Ataman and Alparslan[21] have investigated the effect of those two windows. Fig. 5 a, b,c,d and e show the original 128 x 128 image, original image corrupted by gaussian noise, 3-point median filtered image with square window, 3-point median filtered image with cross window and linear filtered image, respectively.

Narendra[12] has considered processing images by separable filters rather then two-dimensional filters. In the former the pixels in each row of the picture matrix are processed by a one-dimensional window and then the columns are processed by the same window. He shows that the performance, insofar as the smoothing is concerned, of the separable filter is identical to that of a square window filter. He also notes that a separable filter yields a slightly greater variance than its two dimensional counterpart. While a linear smoother performs better for uniform and normal distributions, median filters in general perform better on spiky noise.

In the above equations, L_{SS}, D_{SS} and $P_{SS}(0,t)$ represent steady-state values of $L(t)$, $D(t)$ and $P(0,t)$, and the subscript i is omitted as these relations are valid for each i. An expression which satisfies all three conditions is of the following form:

$$f[L_i(t),D_i(t)] = \frac{1}{1+L_i(t)}\, e^{\gamma[1-\frac{L_i(t)(1+L_i(t))}{D_i(t)}]} \qquad (10)$$

where $0<\gamma<1$. Computations show that the best agreement with exact analytical solutions is obtained with $\gamma = 0.55$, and equations (9) and (10) give

$$U_i(t) = 1- \frac{1}{1+L_i(t)}\, e^{0.55[1- \frac{L_i(t)(1+L_i(t))}{D_i(t)}]} \qquad (11)$$

Customers arrived at the i-th station could be newcomers just enter the network with the entrance rate $\nu_{0i}(t)$ or transfered from other stations with the transfer probability T_{ji} after jobs had been completed there. Thus, the arrival rate $\nu_i(t)$ can be represented by:

$$\nu_i(t) = \sum_{j=1}^{j=S} \mu_j\, U_j(t)\, T_{ji} + \nu_{0i}(t) \qquad (12)$$

To represent changing conditions in the queueing network, the parameters μ_i and T_{ji} can be assigned functions of time t.

MODEL STABILITY

In the following it will be shown that the system of equations (7), (8) and (11), which represents an individual server i, is stable with respect to small perturbations of its trajectory. The arrival rate $\nu_i(t)$ is assumed to be exogenous or known function of t. Our purpose is to show that our representation of each individual server is stable, and consequently any error in the numerical integration process tends to die out instead of growing exponentially.

With a small perturbation in $U_i(t)$, equations (11) gives:

$$\delta U_i(t) = \frac{\delta L_i}{1+L_i}\left[\frac{0.55(1+2L_i)}{D_i} + \frac{1}{1+L_i}\right]\cdot \mathrm{EXP}\left[0.55\left[1- \frac{L_i(1+L_i)}{D_i}\right]\right]$$

$$- \frac{0.55 L_i\, \delta D_i}{D_i^{\,2}}\cdot \mathrm{EXP}\left[0.55\left[1- \frac{L_i(1+L_i)}{D_i}\right]\right]$$

$$= A_i\,\delta L_i - B_i\,\delta D_i \qquad (13)$$

where A_i and B_i are positive, and denote the rational functions of D_i, L_i as shown.

Equation (7) with perturbation becomes

$$\frac{d\delta L_i}{dt} = -\mu_i\, \delta U_i(t) = -\mu_i A_i\,\delta L_i + \mu_i B_i\,\delta D_i \qquad (14)$$

Similarly, equation (8) becomes

$$\frac{d\delta D_i}{dt} = [-2\mu_i(1-U_i(t))+\mu_i(1+2L_i)A_i]\delta L_i - \mu_i(L+2L_i)B_i\,\delta D_i \qquad (15)$$

Since the determinant is

$$\begin{vmatrix} -\mu_i A_i & \mu_i B_i \\ -2\mu_i(1-U_i(t))+\mu_i(1+2L_i)A_i & -\mu_i(1+2L_i)B_i \end{vmatrix}$$

$$= \mu_i^2 A_i B_i(1+2L_i)+2\mu_i^2 B_i(1-U_i(t))-\mu_i^2 A_i B_i(L+2L_i) > 0$$

and diagonal elements are negative, the perturbation equations (14) and (15) represent a stable system.

SIMULATED RESULTS

Figure 1 represents the simulated results of step response of a single server queueing system with input $\nu\tau=0.9$ where τ is the mean service time $1/\mu$. It is noted that the average queue length at $t=18\tau$ has only reached 3.3 or 36.6% of its steady-state value. It is an indication of the time required to reach equilibrium state. We also find that the present simulated results are very close to those obtained by Fan's method.

The present simulation method equations (7), (8) and (11) is not only valid in the case $L(0)=0$ but also valid in the case where the initial queue length $L(0) \neq 0$. Equally good simulated results have been obtained and verified.

Fan's three-station network with sinusoidally varying arrival rate is used as a demonstrable example which is specified by the following parameters:

$\nu_{01}(t)=3.6(1-\mathrm{Cos}\ 2\pi t/24)$, $\nu_{02}(t)=\nu_{03}(t)=0$,

$\mu_1 = 6$, $\mu_2 = \mu_3 = 3$, $T_{12} = 0.4$, $T_{13} = 0.6$,

and all other $T_{ij}=0$. Fan's results are given as continuous curve in Figure 2, while simulated results by the present method are given as dots. The dashed curves are obtained by using steady-state solutions. It is found that simulated results by Fan's and present methods agree very closely. Simulated results by both methods have a substantial difference with those by steady-state solutions.

In his paper, Rider gives an example with varying arrival and service rates and compared his result with that of an exact calculation, and that culculated from the steady-state assumption. His result is demonstrably better than the latter. Using the same arrival and service rates, the average queue length is also obtained by the present method and added to Rider's in figure 3. It is seen that the present result is substantially closer to the exact solution than Rider's.

4. PRIORITY QUEUEING NETWORK WITH PRE-EMPTIVE DISCIPLINE

In a priority queue with pre-emptive resume discipline, a queue of lower priority is transparent to a customer of higher priority. Let there be W classes of customers. Let $L_{ik}(t)$, $D_{ik}(t)$ be the average queue length and variance of class-k customer at the i-th server, respectively. $\nu_{ik}(t)$, $\mu_{ik}(t)$ are arrival and service rates, respectively, of class-k customer at the i-th server. $Q_{ik}(t)$ is the probability that all queues of priority higher than k are empty. $\underline{P}_{ik}(0,t)$ is the probability that all queues of priority higher than k including k are empty. Equations (7) and (8) become:

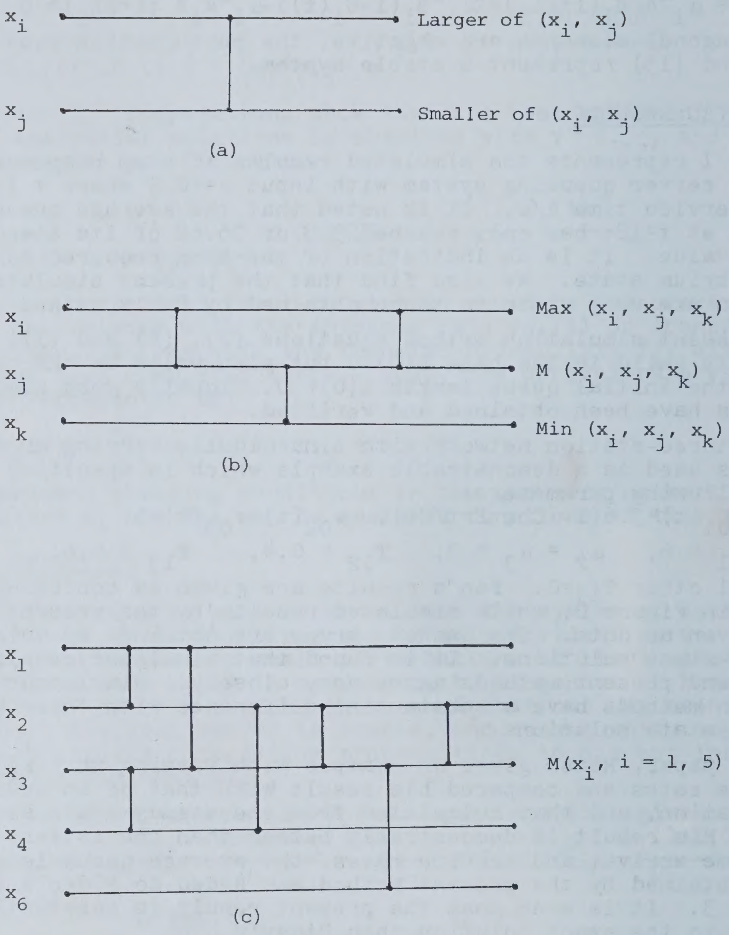

x_i
Larger of (x_i, x_j)
x_j
Smaller of (x_i, x_j)
(a)
x_i
Max (x_i, x_j, x_k)
x_j
M (x_i, x_j, x_k)
x_k
Min (x_i, x_j, x_k)
(b)
x_1
x_2
x_3
M(x_i, i = 1, 5)
x_4
x_6
(c)

Fig. 2

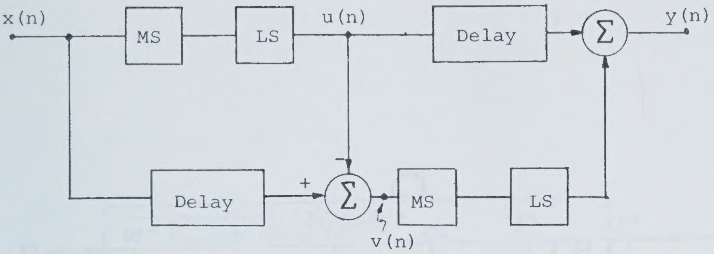
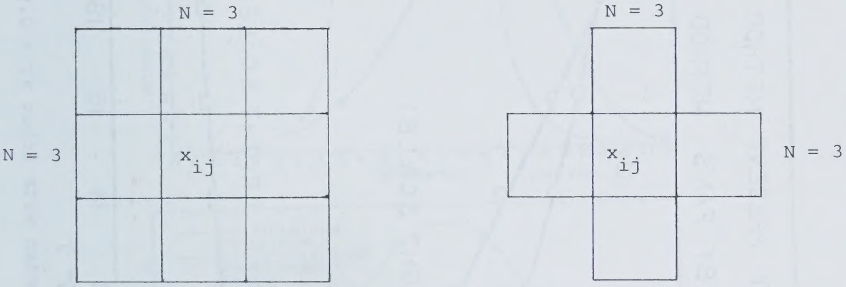

x(n)
MS
LS
u(n)
Delay
Σ
y(n)
Delay
+
−
Σ
MS
LS
v(n)

FIG. 3


N = 3
N = 3
x_{ij}
N = 3
N = 3
x_{ij}

(a)

(b)

Fig. 4

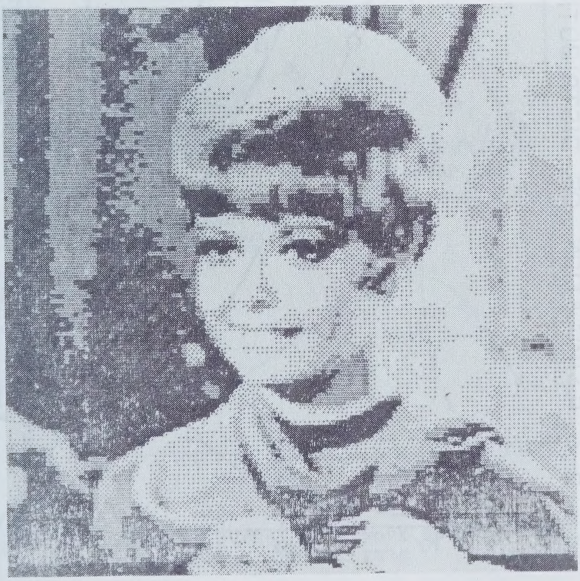


Fig. 5 (a)

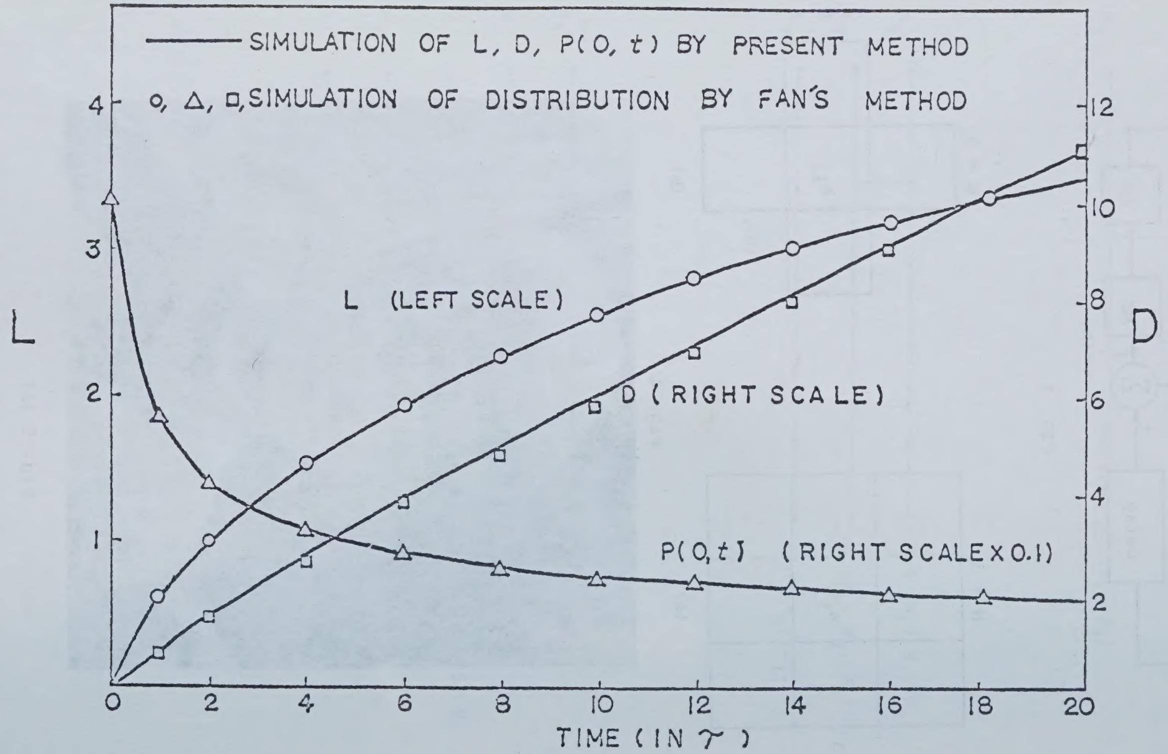
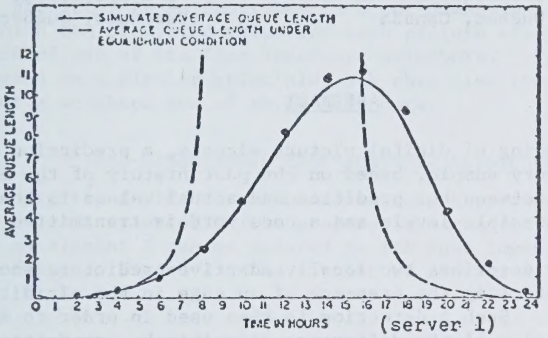
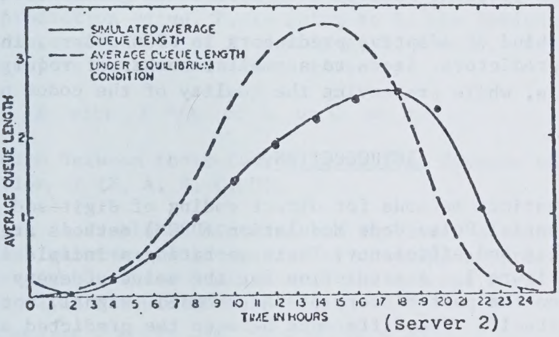


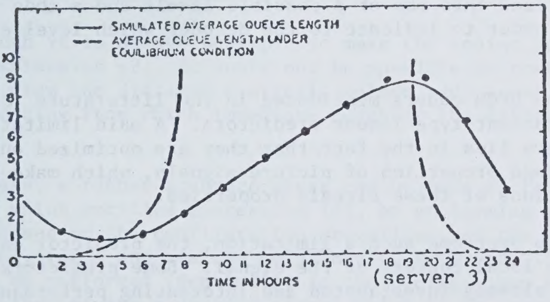
Fig. 1 Step response of a M/M/1 queueing system with input $\nu\tau = 0.9$



(server 1)



(server 2)



(server 3)

Fig. 2. Response of queueing network to periodicaly varying Poisson arrival rate

EDGE-ADAPTIVE PREDICTORS FOR DPCM CODING OF PICTURES

by

P. COHEN
Department of Electrical Engineering
Ecole Polytechnique de Montréal
Montréal, Québec, Canada

A. VALETTE
Institut de Recherche de
l'Hydro-Québec
Varennes, Québec, Canada

ABSTRACT

In DPCM coding of digital picture signals, a prediction is made for the value of every sample, based on the past history of the encoded signal. The difference between the predicted and actual values is then quantized into one of n possible levels and a code word is transmitted accordingly.

This paper describes two locally adaptive predictors whose originality consists of detecting the presence of an edge in the vicinity of every picture element. Such a detection is then used in order to minimize the instantaneous value of the difference signal to be transmitted.

Using this kind of adaptive predictors in DPCM coders, instead of invariant type predictors, leads to a smaller number of required quantizing levels, while preserving the quality of the coded pictures.

INTRODUCTION

Among the various methods for direct coding of digitized picture signals, Differential Pulse Code Modulation (DPCM) methods are characterized by simplicity and efficiency. Their operating principle is illustrated in Figure 1. A prediction for the value of every sample is made, based on the past history of the encoded signal, not the original picture signal itself. The difference between the predicted and actual values is quantized into one of n possible levels and a code word is transmitted in order to indicate to the decoder which level effectively occurred.

Most of the DPCM coders mentionned in the litterature [1,2] make use of invariant-type linear predictors. A main limitation of these predictors lies in the fact that they are optimized only with respect to average properties of picture signals, which make them blind to local variations of these signals properties.

In order to overcome such a limitation, the predictor can be made adaptive to the local context of the signal. Some predictors of this type have been already investigated and interesting performance where obtained.[3]

This paper describes two locally adaptive predictors whose originality consists of observing first the context (i.e. the neighboring elements) of every picture element in order to detect the presence of an edge in the vicinity of this element. Once this detection is accomplished, the predictor chooses,among the neighboring elements, which ones are to be considered in the computation of the predicted value. Such a procedure tends to improve the performance of the predictor by eliminating all the neighbors whose luminance is not close to the luminance of the current element.

The first predictor to be described is a one-dimensional adaptive predictor in which the predicted value for each picture element is equal to the luminance of one of its four immediate neighbors. The second predictor is based on a similar principle, but this time the predicted value is made of a weighted sum of three neighbors.

ONE-DIMENSIONAL EDGE-ADAPTIVE PREDICTOR

Since a digitized picture can be adequately represented in each of its directions by a first-order Markov process, the context to be considered for an element X can be reduced to its four immediate neighbors A, B, C, D according to Figure 2. The context of element A itself consists of the elements E, F, B, C.

The predictor being considered here is such that, at each sampling instant, the prediction value, P, is going to be the luminance of one of the four immediate neighbors of X, namely A, B, C or D. Accordingly, the difference signal to be transmitted is given by:

$$e = X - P \text{ with } P = A \text{ or } B \text{ or } C \text{ or } D \tag{1}$$

where the choice between these four alternatives depends on the particular configuration of (X, A, B, C, D).

Since it is desirable to minimize the amplitute of the difference signal, an ideal solution would consist of choosing as prediction element the immediate neighbor whose luminance is the nearest to X. In this case, the prediction value P would be such that:

$$|X - P| = \min \left[|X - A| , |X - B| , |X - C| , |X - D| \right] \tag{2}$$

However, although it is always possible to make the coding loop operate according to expression (2), it would not be possible to restore uniquely the value of X from the difference signal, unless additional information is given, indicating from which immediate neighbor the difference e was taken.

Nevertheless, a rather accurate guess can be made of the immediate neighbor of X which verifies expression (2), by performing some sort of extrapolation based on the configuration prevailing, at the preceding sampling instant, for A and its own neighbors E, F, B and C. Such a prediction can be performed identically in the coding loop and the decoding loop, thus giving a decoded signal which is an exact replica (ignoring the quantizing noise) of the original signal.

A RESULT IN THE CONTROL OF SYSTEMS GOVERNED BY LINEAR DIFFERENTIAL-DIFFERENCE EQUATIONS

ARILD THOWSEN
School of Electrical Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332

ABSTRACT

The problem of controlling the state of a linear time-invariant differential-difference equation system to the zero state of the appropriate function space is considered with control inputs restricted to piecewise constant functions. Such control inputs frequently occur in sampled-data control of processes. Earlier work revealed that an algebraic necessity condition implied that the characteristic quasipolynomial must be a polynomial for all second and third order systems in which the control objective was achieved. It is shown by spectral analysis arguments that this requirement holds true for linear time-invariant single delay systems of any order.

I. INTRODUCTION

Since some physical and biological systems are governed by differential-difference equations, it may be of interest to determine conditions for bringing the state of such systems to the zero state. It is known [1] that a multiple delay system of the form

$$\dot{y}(t) = \sum_{i=0}^{m} A_i\, y(t-ih) \tag{1}$$

with $y(t)\varepsilon R^p$ and $h>0$ can be transformed to a system with a single unit delay

$$\dot{x}(t) = Ax(t) + Bx(t-1)\ , \quad x(t)\varepsilon R^n \tag{2}$$

where $n=pm$. Therefore this paper will be concerned only with the control of system (2). Control inputs for process (2) are commonly generated based on sampled values of some linear combination of the components of the vector function $x(\cdot)$. The sampling will be assumed to occur at the integer time values $t=0, 1, 2, \ldots,$ and it is further assumed that the control signals remain constant between sampling instants. Note that the state x_t of system (2) at time t is defined as

$$x_t \triangleq x(t+\theta), \quad \theta\,\varepsilon\,[-1, 0]$$

This work was supported by the National Science Foundation under Grant ENG 78 - 12231.

Hence x_t is an element of a function space, e.g. $\mathscr{C}\left([-1,\ 0];\ R^n\right)$. The specific problem considered can then be restated as follows:

Given the controlled process

$$\dot{x}(t) = Ax(t) + Bx(t-1) + u(t), \quad t > 0 \tag{3}$$

with arbitrary initial condition $x(t) = \phi(t),\ \phi \varepsilon \mathscr{C}\left([-1,\ 0];\ R^n\right)$, determine necessary conditions for controlling the system state from $x_0 = \phi$ to the zero state in finite time using piecewise constant control

$$u(t) = u(k), \quad k \leq t < k+1 \tag{4}$$

for $k = 0, 1, 2, \ldots$.

This problem was briefly considered in [2] where it was shown that there exist time-delay systems for which this type of control is possible. Other prior results, applicable to low-order systems, are reviewed in the next section. In Section III the main result is presented along with an outline of the proof. Specifically, theorem 1 states that it is necessary for system (3) to have a finite spectrum in order to be zero state controllable with control (4). The paper concludes in Section IV with a brief discussion of some implications for the design of zero state sampled-data controllers for process (3).

II. CONTROL OF LOW ORDER SYSTEMS

A convenient representation of the infinite dimensional system (3)-(4) consists of a finite dimensional differential equation and an associated boundary condition. This representational form has been frequently used in the study of pointwise degeneracy of time-delay systems [3,4,5]. Define for $p \geq 1$ the matrices

$$A_p \triangleq \begin{bmatrix} A & & & & \\ B & A & & 0 & \\ & B & \cdot & & \\ & & \cdot & \cdot & \\ & 0 & & \cdot & \cdot \\ & & & B & A \end{bmatrix}_{np \times np} \qquad B_p \triangleq \begin{bmatrix} B \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix}_{np \times n}$$

and the n-dimensional vectors

this last case, the difference between the original and decoded values must be taken into account in the coding loop, in order to avoid systematic error propagation.

This operation principle is illustrated by the flow-chart of Figure 4.

Several estimation procedures have been tried for the situations where e_T is not uniquely decodable. Table 2 illustrates comparative results for the most successful procedure. In this table, Predictor No.1 is a fixed two-dimensional predictor (where P is always equal to $A+C-B$) and predictor No.2 is the ideal predictor corresponding to equation (5). Here again, a definite superiority of the proposed predictor can be observed. over the fixed two-dimensional predictor.

CONCLUSION

This paper has presented two edge-adaptive predictors which can be successfully used in DPCM loops for black-and-white picture signals. These predictors show better performance, in term of the difference signal statistics, than invariant-type linear predictors.

Better performance is also obtained in terms of slope-overload noise, because of the local edge detection involved in the adaptation procedure. Consequently, the number of required quantizing levels may be reduced, while preserving the quality of the decoded pictures.

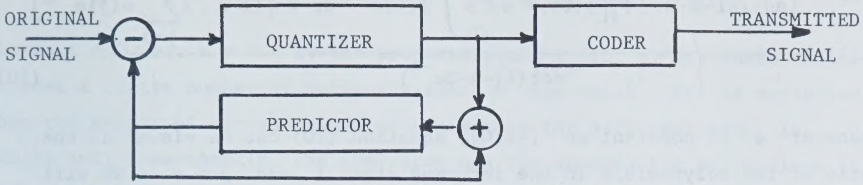
REFERENCES

[1] Connor, D.J., Brainard, R.C., Limb, J.O.: "Intraframe Coding for Picture Transmission" IEEE Proceedings, Special Issue on Digital Picture Processing, Vol. 60, No.7, (July 1972).

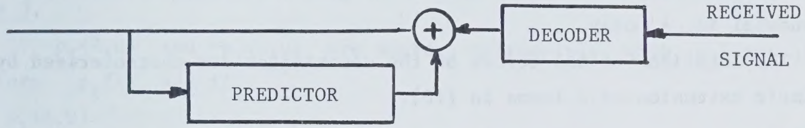
[2] O'Neal, J.B.: "Predictive Quantizing System (Differential Pulse Code Modulation) for the Transmission of Television Signals", Bell System Technical Journal, Vol. 45, (May-June 1966).

[3] Connor, D.J., Pease, R.F.W., Scholes, W.G.: "Television Coding Using Two-Dimensional Spatial Prediction", Bell System Technical Journal, Vol.50, no. 3, (March 1971).

[4] Habibi, A., Robinson, G.S.: "A Survey of Digital Picture Coding", Journal of the IEEE Computer Society, Vol.7, no.5, (May 1974).


ORIGINAL
SIGNAL
QUANTIZER
CODER
TRANSMITTED
SIGNAL
PREDICTOR

(a) CODING LOOP


RECEIVED
SIGNAL
DECODER
PREDICTOR

(b) DECODING LOOP

FIGURE 1 - DPCM CODING SYSTEM

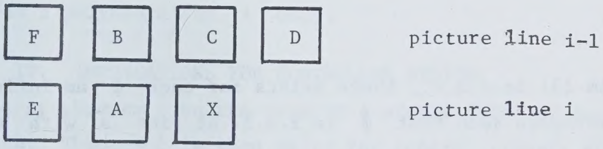

F
B
C
D
picture line i-1
E
A
X
picture line i

Figure 2 - REPRESENTATION OF THE CONTEXT OF ELEMENT X

	Predictor 1 e = X-A	Predictor 2 e = X-A-C+B	Predictor 3 (equ. 2)	Proposed Predictor
Picture 1 (Group)	58.7	25.8	1	23.6
Picture 2 (cameraman)	8.0	8.2	1	4.5
Picture 3 (Portrait)	15.5	11.0	1	4.0

TABLE 1 - ONE-DIMENSIONAL PREDICTOR: MEAN SQUARE VALUE OF THE DIFFERENCE SIGNAL.

$$X(\lambda)= \frac{\left[\mathrm{adj}(\lambda I-A-Be^{-\lambda})\right]\left[\phi(0) + e^{-\lambda}B\int_{-1}^{o}\phi(\tau)e^{-\lambda\tau}dt + \frac{1}{\lambda}(1-e^{-\lambda})\sum_{j=0}^{s-2}u(j)e^{-\lambda j}\right]}{\det(\lambda I-A-Be^{-\lambda})} \tag{10}$$

Whenever ϕ is constant on $[-1,0]$, equation (10) can be viewed as the ratio of two polynomials in the intermediates λ and $\mu \overset{\Delta}{=} e^{-\lambda}$. We will show that the number of zeroes common to both the numerator and the denominator polynomials is finite so that their greatest common divisor is a polynomial in λ only.

The locations of the zeroes of the denominator are characterized by a simple extension of a lemma in [10].

Lemma 1.

There exists a real number α such that all solutions of $\det(\lambda I-A-Be^{-\lambda}) = 0$ satisfy $\mathrm{Re}\,\lambda < \alpha$. Furthermore, there are only a finite number of solutions in any vertical strip in the complex plane.

The asymptotic distribution of the zeroes of the numerator can be determined by dominant term analysis [11] together with the next easily proven lemma.

Lemma 2.

If system (3) is z.s.c., there exists for each ϕ an infinite set S of positive integers such that ϕ is z.s.c. at time s with $u(s-2) \neq 0$, $u(k) = 0 \quad \forall k \geq s-1$, for each $s \varepsilon S$.

With $\phi = 1$ as initial function there exists by lemma 2 at least one component of $u(s-2)$ which is non-zero. Without loss of generality we may take the first component of $u(s-2)$ equal to $a \neq 0$. The dominant terms of the first component of the numerator of (4) for large s as $|\lambda| \to \infty$ and $|\mu| \to \infty$ show that the asymptotic location of the numerator zeroes, $z_k = x_k \pm iy_k$, are determined by solving $\lambda = a\mu^{s-1}$. Hence [11, p.35]

$$x_k(s) = \frac{1}{s-1}\left(-\ln y_k(s) + \ln |a(s)|\right) + \varepsilon$$

where $\varepsilon \to 0$ as $k \to \infty$. By lemma 2 the index set S is infinite. Clearly, if the numerator and the denominator polynomials have an infinite number of common zeroes, these zeroes must be asymptotic zeroes of the numerator and their location in the complex plane must be independent of $s \varepsilon S$. If $s_1, s_2 \varepsilon S$ with $s_1 \neq s_2$ and $y_{k_1}(s_1) = y_{k_2}(s_2) \overset{\Delta}{=} y_k$ for positive integers k_1 and k_2, then $z_{k_1}(s_1) = z_{k_2}(s_2)$ only if

$$(s_1 - s_2)\ln y_k = (s_1 - 1)\ln|a(s_2)| - (s_2 - 1)\ln|a(s_1)| + \hat{\varepsilon} \quad (11)$$

where $\hat{\varepsilon} \to 0$ as $k \to \infty$. By the properties of y_k[11, p. 36] there exists at most a finite number of integer values k for which (11) is satisfied. Then the number of zeroes common to numerators for different $s \in S$ is finite and, consequently, the numerator and the denominator polynomials in (10) do not possess a common divisor $d(\lambda,\mu)$ where d depends explicitly on μ. Lemma 3 from [12] then gives theorem 1 directly.

Lemma 3.

If $p_1(\lambda,\mu)$ and $p_2(\lambda,\mu)$ are nonzero polynomials, with p_2 not of the form $p_2(\lambda)$, and if

$$\frac{p_1(\lambda,\mu)}{p_2(\lambda,\mu)}$$ is an entire function, then the polynomials $p_1(\lambda,\mu)$ and $p_2(\lambda,\mu)$ have a nonconstant greatest divisor $d(\lambda,\mu)$ which is not of the form $d(\lambda)$.

Theorem 1.

If system (3) is z.s.c., then the characteristic quasipolynomial $|\lambda I - A - Be^{-\lambda}|$ is a polynomial in λ only.

IV. IMPLICATIONS FOR CONTROLLER DESIGN

Extension of theorem 1 to the case of a single delay of length h seems immediate. Theorem 1 is then valid for control signals which are piecewise constant over any interval of rational length.

By [13] all second order z.s.c. time-delay systems (3) are pointwise complete since their A and B matrices commute. Consequently, the homogeneous part of (3) in case $n = 2$ is governed by an ordinary differential equation for $t > 1$ [14, 15]. This property, which may also apply to higher order systems, suggests the possibility of designing relatively simple sampled-data feedback controllers for systems (3).

REFERENCES

[1] D. W. Ross, "Controller Design for Time Lag Systems via a Quadratic Criterion," IEEE Trans. Automatic Control, vol. AC-16, no. 6, 1971, pp. 664-672.

[2] A. Thowsen and W. R. Perkins, "On the Controllability of Linear Time-Delay Systems with Piecewise Constant Inputs," Int. J. Sys. Sci., vol. 7, no. 3, 1976, pp. 347-360.

[3] V. M. Popov, "Pointwise Degeneracy of Linear, Time-Invariant, Delay-Differential Equations," J. Differential Equations, vol. 11, no. 3, 1972, pp. 541-561.

SOME EXPERIMENTAL, ANALYTICAL AND COMPUTATIONAL RESULTS CONCERNING MAXIMAL VERTICAL JUMPS

W. S. LEVINE
Dept. of E.E.
Univ. of Md.
College Park, Md.

F. E. ZAJAC*
Rehab. Engrg.
R&D Center
V. A. Hosp.
Palo Alto, CA

M.R. ZOMLEFER
Karolinska Inst.
Physiol. Dept. III
114 33 Stockholm
Sweden

M. R. BELZER
Dept. of E.E.
Univ. of Md.
College Park, Md.

ABSTRACT

There is considerable current interest in the use of optimal control theory to elucidate the techniques whereby animate locomotion is controlled. This paper reports experimental results for humans who were asked to jump as high as possible while keeping their knees and hips locked. The experimental set-up is modelled by a two segment inverted pendulum which is controlled by a single ideal torque generator. The optimal control problem of making this model "jump" to a maximum height is solved. The experimental and analytical results are then compared.

INTRODUCTION

Recently, there have been a number of attempts to apply optimal control theory to the analysis of animal and human locomotion [1], [2], [3]. These attempts have been motivated by problems in prosthesis design, rehabilitation engineering, and sports and by the belief that optimal control theory is a useful technique for the elucidation of complex control problems. There is also interest, at present, in legged vehicles [4] and other anthropomorphic devices [5]. All these problems involve the dynamics of multi-segment pendula. The problems are thus nonlinear, have unusual state constraints and often involve controls that are neither bang-bang nor linear. Because of this, it has generally been impossible to solve these optimization problems analytically. The exception to this has been a recent paper [6] in which the simplest case, that of making a baton "jump" as high as possible, was solved analytically.

The baton problem is too simple to be a reasonable model of any physiological control problem. The next simplest theoretical problem is to make a two segment inverted pendulum, controlled by a single ideal torque generator, "jump" as high as possible. It is possible to experimentally approximate this theoretical problem by asking humans to jump as high as possible while keeping their knee and hip joints locked. This paper compares the results of just such an experiment with the results of an analysis of the two segment, ideal torque controlled, maximum height "jumper".

The results of this study of the two segment or "foot jump" give useful information about the maximum torque which is generated by the triceps surae (calf) muscle group in humans and about physiological bang-bang controls. However, the major reason for studying the foot jump was to

---

Acknowledgement:
This research has been supported in part by N.I.H. Grant #NS11971, in part by Swedish Medical Research Council Projects No. 3026 and 5402 to Professor Sten Grillner, and in part by the Computer Science Center of the University of Maryland. The experimental research was carried out at Karolinska Institute, in Professor S. Grillner's Laboratory, where Dr. Zajac was a visiting scientist from August '78 to June '79.

*Formerly Associate Professor, Department of E. E., University of Maryland.

develop and sharpen the tools needed for the experimental and theoretical study of normal maximum height jumps by both humans and cats. The reasons for our interest in maximum height jumps are outlined in detail in [3]. Briefly, there are four reasons. First, jumping is closely related to running (or galloping in cats). Second, the maximum height vertical jump has a fairly unambiguous performance measure, thereby, simplifying both the theoretical analysis and experimental procedures. Third, the task is done in similar ways by cats and humans, thereby, facilitating comparative studies. Fourth, the maximal vertical jump is more amenable to analysis than more complex tasks such as walking.

The report of our study of the foot jump begins, in the following section, with a careful statement of the theoretical problem. This is followed by the solution to the problem, obtained by a combination of theory and computation. The experimental results are then described. The paper concludes with a comparison of the experimental observations and the analytical predictions.

PROBLEM STATEMENT

Consider the two segment inverted pendulum illustrated in Fig. 1. Assume the pendulum is not fastened to the ground so that a "jump" is possible. For simplicity, assume that the pendulum cannot slide horizontally along the ground. Thus, horizontal forces can be ignored. Otherwise, the masses, lengths and inertias will be chosen to approximate those of the experimental subject. The controller, $u(t)$, is assumed to be an ideal torque generator capable of producing torques bounded by $0 \leq u(t) \leq u_{max}$, where u_{max} corresponds to the maximum ankle torque which the experimental subject could produce.

Given this physical structure, the problem is to find the $u(t)$ $(0 \leq t \leq T)$ which will cause this mechanism to "jump" to the maximum possible height. This must be accomplished without violating either of the constraints (1) neither segment can go through the ground and (2) $-\theta \leq \phi \leq \pi + \theta$.

It is not trivial to translate the above physical problem into a meaningful mathematical optimization problem. The mathematical problem requires four unambiguous components: (a) dynamics, (b) initial and terminal conditions, (c) constraints and (d) performance criterion. It will be seen that there is considerable interaction among (a), (b) and (c).

In particular, consider the dynamics. There are two distinct cases that are important. The first case is illustrated in Fig. 1 and corresponds to only the toe on the ground. The second case corresponds to the whole foot on the ground. One might think it is also necessary to consider the situation where the entire device is airborne. It will be assumed that the objective is to maximize the peak height achieved by the center of mass of the object. This height is completely determined at the instant of lift-off. Hence, there is no need to consider the dynamics once the object is airborne.

Case 1: Only the "toe" on the ground. The dynamics are given by

$$[I_1 + m_1 \ell_1^2 + m_2(\ell_1 + \ell_2)^2]\ddot{\theta} + m_2(\ell_1 + \ell_2)\ell_3 \cos(\theta + \phi)\ddot{\phi}$$

$$- m_2(\ell_1 + \ell_2)\ell_3 \sin(\theta + \phi)\dot{\phi}^2$$

$$+ (m_1\ell_1 + m_2(\ell_1 + \ell_2))\, g \cos\theta = u(t) \qquad (1)$$

E5: If λ_i is given and $|y_i - a(z_i)| < \delta$, we have the case of round-off or data errors.

E6: Let $A(\cdot)$ be a point-to-set map; i.e. for any z in T, $A(z)$ is a nonempty subset of T [6,7]. If y_i belongs to $A(z_i)$ and $\lambda_i = 1$, then we have the Picard iteration for point-to-set maps.

A crucial point to be noted is that the implementation of the proposed model also depends on the manner by which $\{\lambda_i\}$ is determined: (i) The sequence $\{\lambda_i\}$ may be chosen a priori, as in examples E1 - E6 above, or as is done for the one-dimensional Kalman filter, which has the form of Equation (1) for T in E^1. (ii) Alternatively the sequence $\{\lambda_i\}$ may be chosen adaptively at each iteration according to a particular set of problem related criteria [3,4,14]. An example of this type of choice is:

E7: The steepest descent algorithm for minimizing $f(\cdot): T \to E$:

$$z_{i+1} = z_i - \lambda_i \nabla f(z_i) \tag{5}$$

where λ_i is chosen to minimize $f(z_i + \lambda \nabla f(z_i))$ with respect to λ [14,18].

Our results are presented in three sections. First we analyze the properties of the sequence $\{\lambda_i\}$ which are necessary to assure useful relationships between the cluster point or limit point sets of the sequences $\{y_i\}$ and $\{z_i\}$. The main result from this section is:

Theorem 1: Assume that $\{z_i\}$ is generated according to Equation (1). If the sequence $\{y_i\}$ is bounded and $\sum_{i=0}^{\infty} \lambda_i$ diverges, then the cluster point set of $\{z_i\}$ is contained in the convex hull of the cluster point set of $\{y_i\}$.

A direct consequence is that if $\{y_i\}$ converges to y_* and $\sum_{i=0}^{\infty} \lambda_i$ diverges, then $\{z_i\}$ converges to y_*.

The next set of results provides conditions on the sequences $\{\lambda_i\}$ and $\{y_i\}$ that assure the convergence of $\{z_i\}$. By progressively weakening the assumptions on $\{y_i\}$ while strengthening the assumptions on $\{z_i\}$, we are able to uncover the roles played by the various properties of $\{\lambda_i\}$.

In the last section we assume that $\{y_i\}$ is corrupted by bounded errors v_i, i.e. only the sequence $\{y_i + v_i\}$ is known. We derive conditions under which $\{z_i\}$ converges to a neighborhood of the solution and provide an index of algorithm/problem sensitivity by finding the relative size of this neighborhood.

REFERENCES

[1] Browder, F.E., and Petryshyn, W.V., Construction of Fixed Points of Nonlinear Mappings in Hilbert Space, Journal of Mathematical Analysis and Applications, Vol. 20 (1967), pp. 197-228.

[2] Dotson, W.G., On the Mann Iterative Process, Transactions of the American Mathematical Society, Vol. 149 (1970), pp. 65-73.

[3] Dunn, J.C., On Recursive Averaging Processes and Hilbert Space Extensions of the Contraction Mapping Principle, Journal of the Franklin Institute, Vol. 295 (1973), pp. 117-133.

[4] Dunn, J.C., A Simple Averaging Process for Approximating the Solution of Certain Optimal Control Problems, Journal of Mathematical Analysis and Applications, Vol. 48 (1974), pp. 875-894.

[5] Edelstein, M., A Remark on a Theorem of M.A. Krasnoselski, The American Mathematical Monthly, Vol. 73 (1966), pp. 509-510.

[6] Hogan, W.W., Point-to-Set Maps in Mathematical Programming, SIAM Review, Vol. 15 (1973), pp. 591-603.

[7] Huard, P., Optimization Algorithms and Point-to-Set Maps, Mathematical Programming, Vol. 8 (1975), pp. 308-331.

[8] Koliha, J.J., The Solution of Linear Equations in Normed Spaces by Averaging Iteration, SIAM Journal on Mathematical Aanlysis, Vol. 5 (1974), pp. 178-186.

[9] Krasnoselski, M.A., Two Remarks on the Method of Successive Approximations, Uspekhi Matematicheskikh Nauk, Vol. 10 (1955), pp. 123-127.

[10] Kushner, H.J., Stochastic Approximation Algorithms for the Local Optimization of Functions with Nonunique Stationary Points, IEEE Transactions on Automatic Control, Vol. AC-17 (1972), pp. 646-654.

[11] Kushner, H.J., and Clark, D.S., Stochastic Approximation Methods for Constrained and Unconstrained Systems, Springer-Verlag, New York, New York, 1978.

[12] Kushner, H.J., and Gavin, T., Stochastic Approximation Type Methods for Constrained Systems: Algorithms and Numerical Results, IEEE Transactions on Automatic Control, Vol. AC-19 (1974), pp. 349-357.

[13] Ljung, L., Analysis of Recursive Stochastic Algorithms, IEEE Transactions on Automatic Control, Vol. AC-22 (1977), pp. 551-575.

[14] Luenberger, D.G., Introduction to Linear and Nonlinear Programming, Addison-Wesley, Reading, Massachusetts, 1973.

[15] Mann, W.R., Mean Value Methods in Iteration, Proceedings of the American Mathematical Society, Vol. 4 (1953), pp. 506-510.

[16] Ortega, J.M., and Rheinboldt, W.C., Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York, New York, 1970.

[17] Picard, Emile M., Sur l'Application des Methodes d'Approximations Successives a l'Etude de Certaines Equations Differentielles Ordinaires, Journal de Mathematiques, 4 Serie, Tome IX., Fasc. III, 1893, pp. 217-271.

[18] Polak, E., Computational Methods in Optimization: A Unified Approach, Academic Press, New York, New York, 1971.

[19] Robbins, H., and Monro, S., A Stocastic Approximation Method, The Annals of Mathematical Statistics, Vol. 22 (1951), pp. 400-407.

$$\begin{aligned}\frac{dJ(t_f)}{dt_f} &= mg\dot{y}_c(t_f)+m\ddot{y}_c(t_f)\dot{y}_c(t_f)\\ &=\dot{y}_c(t_f)[mg+m\ddot{y}_c(t_f)]\\ &=\dot{y}_c(t_f)F_v(t_f)\end{aligned}$$

where (3) is used to simplify the expression. It is obvious that $\dot{y}_c(t_f)>0$ for an optimal jump and that $F_v\geq 0$ for the "toe" to be on the ground. Thus, $J(t_f)$ has an extremum when $\frac{dJ(t_f)}{dt_f}=0$ and is monotone increasing prior to t_f. Thus, in any state for which it is possible to make $F_v(\theta,\dot{\theta},\varphi,\dot{\varphi},u)>0$ it is advantageous to do so. Therefore, all optimal jumps occur on the curve in the state space corresponding to

$$\max_{0\leq u\leq u_{max}} F_v(\theta,\dot{\theta},\varphi,\dot{\varphi},u)=0 \tag{14}$$

Substituting (1) and (2) into (4) shows that F_v is linear in u. It is possible to show that the coefficient of u, which depends on θ and φ, is positive in most of the state space. It is easy to see that a sufficient condition for this coefficient to be positive is that $0\leq\theta,\varphi\leq\pi/2$ and $(\theta+\varphi)\geq\pi/2$. This region includes all of the possible states at which an optimal jump might leave the ground. Thus, (9) is in fact the terminal condition for an optimal jump and $u(t_f)=u_{max}$.

The next step in deriving the optimal control is to extend the above result backwards in time from the instant of lift-off. To do this it is convenient to formulate a new optimal control problem that is valid only in a subset of the state space. In particular, assume that once the jumper's heel leaves the ground it is not advantageous to return the heel to the ground before lift-off. While this is certainly not true for normal jumps it does appear to be true for the constrained jumps analyzed here. Futhermore, in the case of the normal jump there is experimental evidence that energy storage mechanisms, not included in our model, are the source of the advantage.

Having made the above assumption, we now temporarily restrict our attention to states in which the heel is off the ground ($\theta>0$). For these states the dynamics are given by (1) and (2). The performance criterion is still given by (13), the terminal condition by (9) and the control constraint by (10). The mixed constraint given by (11) still applies and greatly complicates the application of the necessary conditions of the minimum principle. Furthermore, we are fairly sure that (11) is not an active constraint just prior to lift-off. Thus, we apply the minimum principle to the "relaxed" problem obtained by ignoring the constraint (11). As long as the optimal control for this "relaxed" problem does not violate the constraint (11) we know it is the optimal control for the true problem as well.

The necessary conditions then become, besides (1) and (2) in state space form, that

$$H(p_1^*(t),p_2^*(t),p_3^*(t),p_4^*(t),\theta^*(t),\phi^*(t),\dot{\theta}^*(t),\dot{\phi}^*(t),u^*(t))=0 \tag{15}$$

$$\dot{\underline{p}}^*(t)=\left.\frac{-\partial H}{\partial \underline{x}}\right|_* \tag{16}$$

with the condition that

$$u^*(t) \text{ maximizes}_{0\leq u\leq u_{max}} H(\underline{p}^*,\underline{x}^*,u) \tag{17}$$

and the boundary condition

$$\underline{p}^*(t_f) = \left.\frac{\partial J}{\partial \underline{x}}\right|_* + \alpha \frac{\partial Fv}{\partial \underline{x}} (\underline{x}^*(t_f), u_{max}) \tag{18}$$

when

$$\underline{x} = [\theta\ \dot{\theta}\ \varphi\ \dot{\varphi}\ ]^T$$

and α is a constant, to be determined.

We have not been able to find an explicit solution for the necessary conditions. However, there is a straightforward technique for using the computer to obtain a solution. The technique is as follows:

(1) Pick any point $\underline{x}$ satisfying (9). Such an $\underline{x}$ is the terminal state of an optimal jump.

(2) Determine $\underline{p}^*(t_f)$, to within a constant α, from (18).

(3) Determine α from (15).

(4) Integrate (1) and (2) and (16) backwards, with $u(t)=u_{max}$ while checking (17) to ensure that $u^*(t)=u_{max}$

(5) The first state at which either u_{max} does not satisfy (17) or $\theta=0$ is on the boundary of the region where $u^*(t)=u_{max}$.

The above calculation can be repeated until the boundary of the region where $u^*(t)=u_{max}$ is established. We have not yet done this. However, we have done the above calculation from our candidate optimal terminal state to verify its optimality. In that case, $u(t)=u_{max}$ all the way back to $\theta=0$ (heel on the ground). It is conjectured, but is not yet proven, that the same result ($u(t)=u_{max}$ once heel-off occurs) is valid for all jumps that are close to optimal.

The last step in deriving the optimal control is to compute the set of states that can be reached from the initial state ($\theta=\dot{\theta}=\dot{\varphi}=0, \varphi=90^{\circ}$) while keeping the heel on the ground. This set is shown in Fig. 2. The derivation of Fig. 2 is based on (5)-(8), the equations describing the dynamics when the heel is on the ground. The crucial requirement is that ℓ_{cp} (see (8)) remain greater than or equal to zero so the heel stays on the ground. As long as the heel is on the ground the dynamics are given by (5) and, as shown in [6], the phase plane trajectories can be calculated analytically.

In deriving Fig. 2 it is convenient to start with the effect of the constraint

$$\ell_{cp}(\theta,\dot{\theta},\varphi,\dot{\varphi},u) = \ell_{cp}(0,0,\varphi,\dot{\varphi},u) \geq 0 \tag{19}$$

In region I, at the bottom of the figure, it is impossible for (19) to be satisfied. Basically, the centrifugal force is large enough to pull the heel off the ground independent of the control u. The boundary of region I is given by

$$\ell_{cp}(0,0,\varphi,\dot{\varphi},0) = 0 \tag{20}$$

The second effect of the constraint (19) is illustrated by the roughly elliptical curve in Fig. 2. In the region exterior to this curve, the heel will leave the ground provided the control u(t) satisfies

$$u(t) > u_a$$

The curve itself satisfies

$$\ell_{cp}(0,0,\varphi,\dot{\varphi},u_a) = 0$$

only be in mode 2, 3, or 4, the number of control states is $C_1(n)=3$.

IV MODEL 2

This model consists of n loops, each of size 1. Between loop i and loop (i+1), there is a switch s_i, for i= 1, 2, · · · , n-1. Thus there are a total of (n-1) switches.

```
procedure sort_2(n, type);
(comment: type 1 means the sorted keys are in ascending order starting from
  loop 1; type 2 means the sorted keys are in descending order starting from
  loop 1);
variable switch: array [1. . . n];
begin
(comment: initially the beginning of the records in loop i and in loop (i+1)
  are at switch i, for i odd);
for i ← 1 to n do
    if odd(i) then switch [i] ← type + 2
               else switch [i] ← 1;
move (1/2);
for i ← 1 to $\lfloor \frac{n-1}{2} \rfloor$ do
    switch [2*i] ← type + 2;
move ($\lfloor \frac{n-1}{2} \rfloor + \frac{1}{2}$);
end;
```

In general, we have the number of steps: $S_2(n)=\lceil n/2 \rceil$, the number of switches: $W_2(n)=n-1$, and the number of control states: $C_2(n)=3$.

V. MODEL 3

This model uses the concept of bitonic sorting [1].

Definition. Let $a_1 \leq a_2 \leq \cdots \leq a_{k_1}$, $b_1 \leq b_2 \leq \cdots \leq b_{k_2}$, $k_1 + k_2 = n$. Then the sequence (or any cyclic shift thereof) $a_1 a_2 \cdots a_{k_1} b_{k_2} b_{k_2-1} \cdots b_2 b_1$ is called a bitonic sequence.

Definition. The process of sorting a bitonic sequence in order is called a bitonic sort.

Theorem 1. If $a_1 a_2 \ldots a_{n-1} a_n$ (n even) is a bitonic sequence, then the sequences

(1) $\min(a_1, a_{\frac{n}{2}+1}),\ \min(a_2, a_{\frac{n}{2}+2}), \ldots,\ \min(a_{\frac{n}{2}}, a_n)$

(2) $\max(a_1, a_{\frac{n}{2}+1}),\ \max(a_2, a_{\frac{n}{2}+2}), \ldots,\ \max(a_{\frac{n}{2}}, a_n)$

are also bitonic sequences, and all the numbers in (1) are no larger than any number in (2).

Theorem 2.

$$\text{Bitonic sort}\,[a_1 a_2 \ldots a_{n-1} a_n] = \text{Bitonic sort}\,[\min(a_1, a_{\frac{n}{2}+1}), \ldots, \min(a_{\frac{n}{2}}, a_n)],$$

$$\text{Bitonic sort}\,[\max(a_1, a_{\frac{n}{2}+1}), \ldots, \max(a_{\frac{n}{2}}, a_n)]$$

In model 3, there are m loops each of size h(n=hm). The loopse are numbered 1, 2, · · ·, m. We assume that m is a power of 2. There is a switch s_i between loop i and loop (i+1) for i=1 to n-1. Each loop is of the structure of model 1, i.e. each loop consists of two smaller loops and a switch between them. One of these two loops is of size 1 and is called the "head" of the whole loop. The switch is called t_i if the loop is numbered i. For i=1, 2, · · ·, n/2, the heads of loops (2i-1) and 2i are placed next to the switch s_{2i-1}.

```
procedure shift (loop_number, size, number_steps);
(comment: suppose loop_number = p, size = q. Assume the switches have
been set properly so that loop p, loop (p+1), . . . , loop (p+(q/h)-1) form a
large loop of size q. This procedure shifts the large loop for "number_steps"
record lengths);
begin
    for i ← loop_number to (loop_number + (size)/h-2) do
       begin
       s[i] ← 2;
       t[i] ← 2;
       end;
    t[loop_number + (size/h-1] ← 2;
    move (number_steps);
end;
procedure sort_1a (loop_number, type);
begin
    for i ← 1 to (h-1) do
       begin t[loop_number] ← type + 2;
            move (h-1);
            t[loop_number] ← 2;
            move;
       end;
end;

procedure bitonic_split (loop_number, size, type: inter);
 (comment: This procedure splits a given bitonic sequence into 2 according
  to Theorem 1. The types are shown in Fig. 2(a));
begin
    for i ← loop_number to (loop_number + size/h-2) do
    begin
       s[i] ← 2;
       t[i] ← 2;
    end;
t[loop_number + size/h-1] ← 2;
s[loop_number + size/(2*h)-1] ← type + 2;
move(size/2);
end;

procedure bitonic sort (loop_number, size, type: integer);
  (comment: given a bitonic sequence of size "size" starting from loop "loop_
   number", sort according to type 1, 2, 3, or 4 as shown in Fig. 2(b));
```

D.R. Humphrey, Raven Press, New York, 1979.

4. McGhee, R.B., "Robot Locomotion", Proceedings of the Int'l. Conf. on Neural Control of Locomotion, Phila., Pa. 1975.
5. Stepanenko, Yu, and Vukobratovic, M., "Dynamics of Articulated Open-Chain Active Mechanisms", Mathematical Biosciences, 28, 1976, pp. 137-170.
6. Roberts, W.M., Levine, W.S., and Zajac, F.E., "A Model for Maximum Height Jumping", (to appear) IEEE Trans. on AC, Vol. AC-24, No. 5, October, 1979.
7. Cho, Y.M., "Optimal Control of Two Segment Limb Model", M.S. Thesis, Univ. of Maryland, Dept. of E.E., 1978.

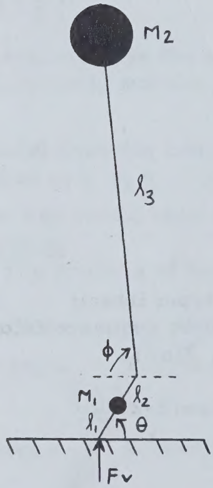


Fig. 1 - Mathematical model and notation

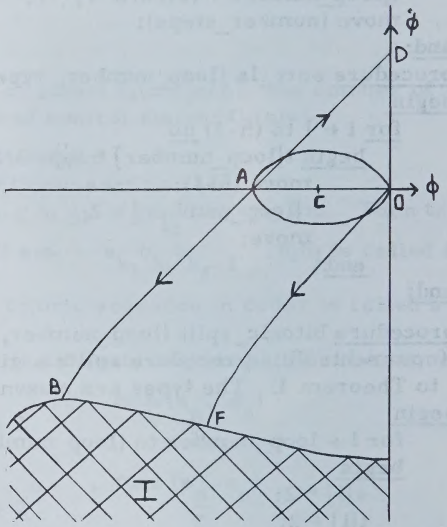


Fig. 2 - State space and reachable states with heel on the ground

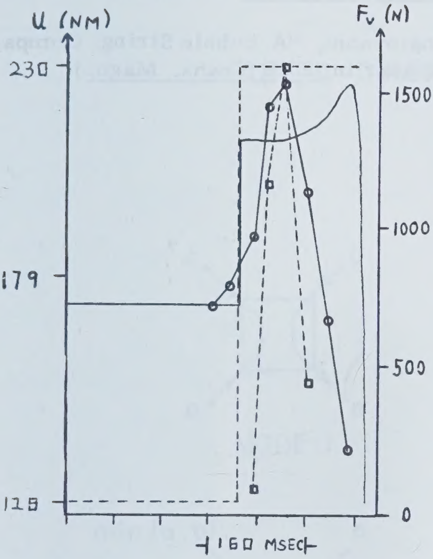


Fig. 3 - Torque and vertical force versus time

— F_v theoretical
-⊖- F_v experimental
--- u^* theoretical
-⊡- u^* experimental

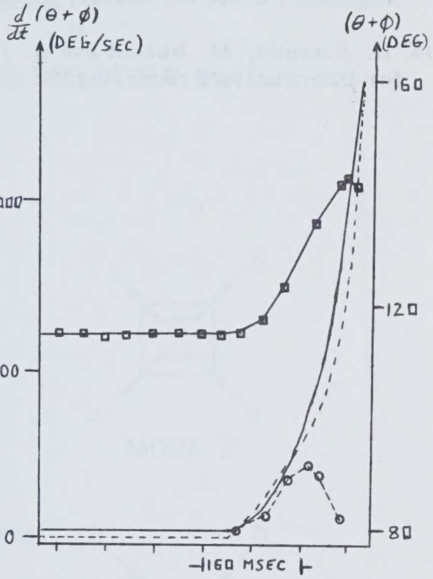


Fig. 4 - Ankle angle and its derivative versus time

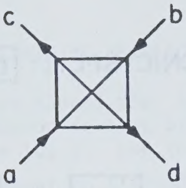
— $\theta+\phi$ theoretical
-⊡- $\theta+\phi$ experimental
--- $\dot{\theta}+\dot{\phi}$ theoretical
-⊖- $\dot{\theta}+\dot{\phi}$ experimental

11. K.M. Chung, F. Luccio and C.K. Wong, "On the Complexity of Permuting Data in Magnetic Bubble Memory Systems," IBM Research Report, RC. 7632, 1979, (also to appear in IBM J. Res. Develop.)

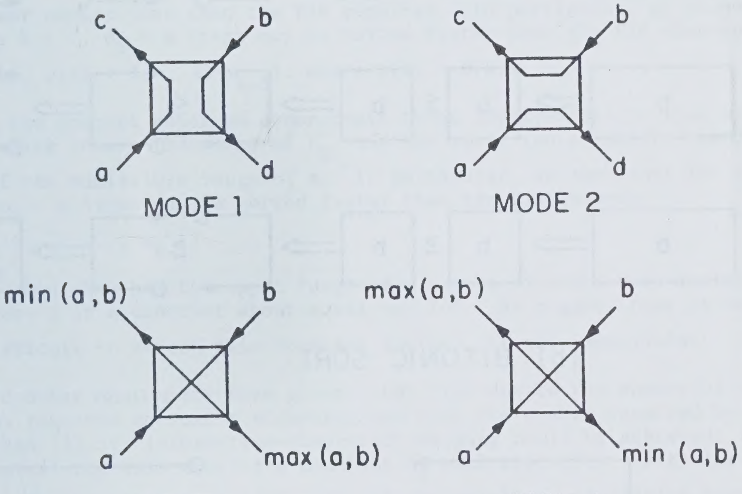
12. C. Tung, T.C. Chen, and H. Chang, "A Bubble Ladder Structure for Information Processing," IEEE Trans. Magn. MAG - 11, 1163-1165 (1975).

13. C.K. Wong and D. Coppersmith, "The Generation of Permutations in Magnetic Bubble Memories, IEEE Trans. on Comp. C-25, 254-262 (1976).

14. B. Antonini, M. Bacchi and G. Bongiovanni, "A Bubble String Comparator for Information Processing," (to appear in IEEE Trans. Magn.)



(a) A COMPARE AND STEER SWITCH

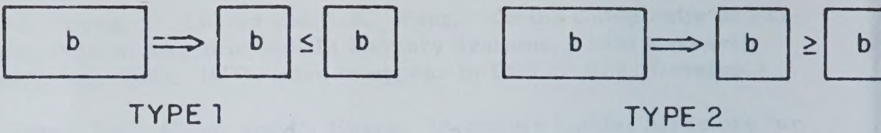

c
b
a
d
c
b
a
d
min (a,b)
b
a
max (a,b)
max (a,b)
b
a
min (a,b)

MODE 1 MODE 2

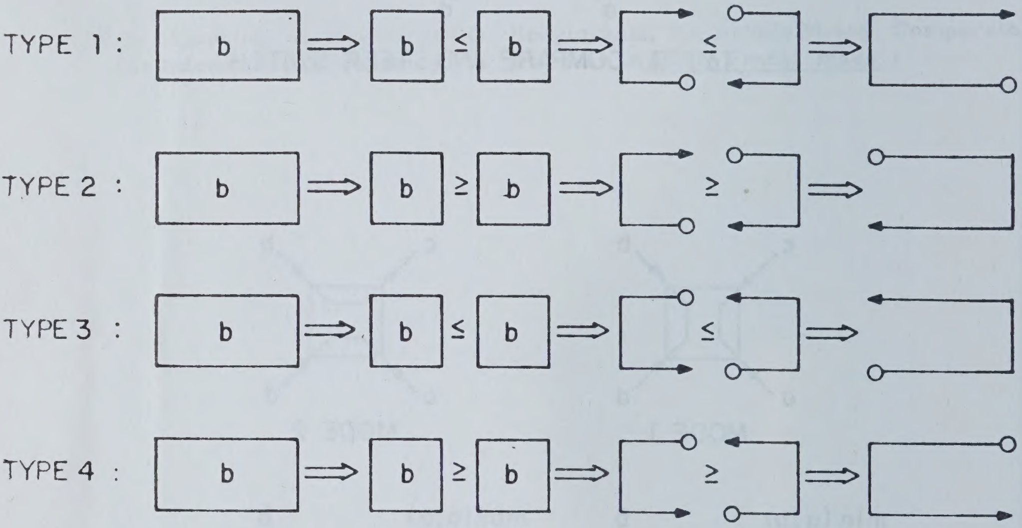
MODE 3 MODE 4

(b) THE FOUR MODES OF A SWITCH

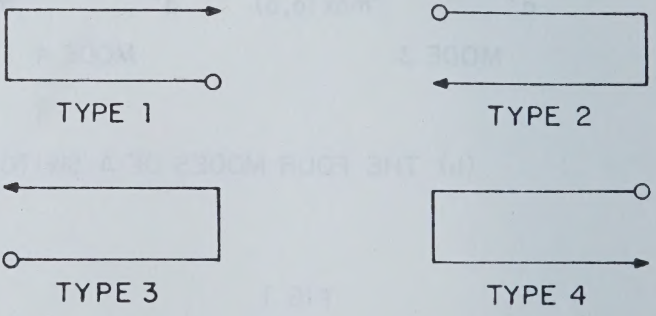
FIG. 1


b
b
≤
b
TYPE 1
b
b
≥
b
TYPE 2

(a) BITONIC SPLIT: $\boxed{b}$ MEANS BITONIC SEQUENCE


TYPE 1 :
b
b
≤
b
≤
TYPE 2 :
b
b
≥
b
≥
TYPE 3 :
b
b
≤
b
≤
TYPE 4 :
b
b
≥
b
≥

(b) BITONIC SORT


TYPE 1
TYPE 2
TYPE 3
TYPE 4

(c) SORTED ORDER IN MODEL 3

FIG. 2

FURTHER RESULTS ON NEAR-OPTIMAL SORTING

GLENN K. MANACHER
University of Illinois, Chicago Circle
Chicago, Illinois 60680

PRELIMINARY ABSTRACT

In a previous paper, [1] we showed that in terms of the number of comparisons required, the Ford-Johnson sorting algorithm may be surpassed in "bands" defined periodically in log t. Let the Ford-Johnson algorithm (FJA) sort t items. Let t be equal to $\lfloor (4/3)2^{k+f} \rfloor$, where k is an integer and $0 < f < 1$. The values $u_k \equiv \lfloor (4/3)2^k \rfloor$ are where the FJA "does best," in the sense that it comes very close to the information-theoretic lower bound for sorting; in fact, no method has ever been found for surpassing the FJA when $t = u_k$. In the same paper, we showed that for all $k \geq k_0 = 7$, there exists a contiguous band of integers lying within a range R defined by $0 < f < \log_2(9/8) = .17$, such that all integers inside the band may be sorted with fewer comparisons than the FJA requires. In particular, we showed that for $k \geq 7$, $u_k + m$ items may be sorted faster than the FJA whenever $m \in \{b(k), b(k) + 1, \ldots, u_{k-3}\}$, where $b(k) = O(k)$.

In the present paper we demonstrate three improvements to this result, one of which lowers the value of k_0, and the other two of which extend both sides of the admissible range of m. In particular, we show that for all $k \geq 5$, $u_k + m$ items may be sorted faster than the FJA whenever $m \in \{5, 6, 7, \ldots, u_{k-1}\}$.

We then show how the upper range of m may be extended even further to zu_k, where z is a constant about equal to .760. We suggest that it may be very difficult to extend this constant further, if not impossible.

Two other results are then given. Let S(t) denote the number of comparisons required to sort t elements, and F(t) the number required by the FJA. Then (1) if "information-theoretic" merging could be achieved, it would entail the existence of a constant t_0 such that $S(t) < F(t)$ for all $t > t_0$. We show that $t_0 \leq 1365$. (2) If the existing information-theoretic lower bound for merging is raised very slightly, in fact by a term proportional to the logarithm of the smaller of the lists being merged, it appears to follow that no combination whatsoever of sorting and merging can beat the FJA when $t = u_k$.

INTRODUCTION

Let $u_k = \lfloor (4/3)2^k \rfloor$. When $t = u_k$, the FJA requires $F(t) = t \log t - 1.415t + O(\log t)$ comparisons,[†] remarkably close to the information-theoretic lower bound for sorting of $t \log t - 1.442t + O(\log t)$. No algorithm is known that will surpass the FJA (in comparisons performed) when $t = u_k$. For others values of t, the present author [1] showed that the FJA can be surpassed. The key idea is that if two sets of size u_k and u_{k-d}, $d \geq 1$ are sorted using the FJA, then merged using the merge-insertion algorithm of Hwang and Lin [3], the result still does not improve on $F(u_k + u_{k-d})$, but the difference $F(u_k + u_{k-d}) - (F(u_k) + F(u_{k-d}) + HL(u_k, u_{k-d}))$, while negative, is only of order log t, where $t = u_k + u_{k-d}$ and HL(m,n) is the number of comparisons required by the Hwang-Lin algorithm (HLA). Therefore any improvement over the HLA that beats it by a term proportional to t will beat the FJA. In [4] we demonstrated such an improved algorithm, and in [1] showed that it led to an improvement to the FJA for t as small as 189.

[†] All logarithms are to the base 2.

Let an arbitrary t be represented by

$$t = (4/3)2^{k+f} \tag{1}$$

where k is an integer and $0 < f < 1$. Then [1,2]

$$F(t) = t \log t - t(3 - \log 3) + t\,\theta(f) + \tfrac{1}{2} \log t + O(1) \tag{2}$$

where $3 - \log 3 \simeq 1.415$, and

$$\theta(f) = 2 - f - 2^{1-f} \tag{3}$$

is a function that is zero at $f = 0$ and 1, and rises almost symmetrically to a value of $(1 - \log e + \log \log e) \simeq .086$ at $f \simeq .47$.

Let all t satisfying $u_k \leq t < u_{k+1}$ be called the k^{th} band. In [1], we showed that the existence of a sorting method involving merging u_k elements with u_{k-d} that beats the FJA entailed the existence of a merging method that would result in bettering the FJA for all m in a range from a very low value up to u_{k-d}. In fact, we showed

Theorem 1 [1]: For all $k \geq 7$, $u_k + m$ elements may be sorted faster than the FJA whenever $b(k) \leq m \leq u_{k-3}$, where $b(k) = O(k)$. []

The purpose of our work is in part to show that the upper and lower bounds of theorem 1 may be sharply extended and that "$k \geq 7$" may be replaced with "$k \geq 5$." We then show that the methods that produced these results do not lead to a method for improving the FJA over an entire band.

This work is based strongly on two recent important papers which strongly dominate reference [4]. The first is a paper of Christen's [5], in which he provides both a far more powerful merging method than that of [4], together with an improved version of the "Oracle" method [2] whereby a sharper lower on M(m,n), the number of comparisons necessary to merge m elements with n, $m \leq n$, is obtained. We now review the main results of that paper. He first establishes a merging method, called "forward probe, backward insertion" which, for all $n \geq 4m - 3$, surpasses the HLA by $\lfloor (m - 1)/4 \rfloor$ comparisons (the best achieved in [4] was about m/12). In addition, for $4^j m \leq n \leq 4^{j+1} m$, Christen achieves an improvement of about $m \sum_{i=1}^{j} 4^{-i}$ comparisons, so that for very large n/m the improvement approaches m/3. Second, he shows that his method is optimal for $5m - 3 \le n \le 7m$, a remarkable result. Thirdly, for $3m \le n \le 4m - 3$, he shows that improvement is still possible, going like $m(\frac{1}{4}(n/m) - 3/4)$, ceasing at 3m. In a letter to the author, [6] Christen anticipated some of the results in the first part of this paper.

The second paper, by Schulte Mönting, derives closed formulas for M(4,n) and probably M(5,n) for all n (the values for M(3,n) and M(2,n) are known from earlier work [7,8,10,11]). The formula for M(5,n) is an extremely sharp upper bound, which is probably exact. Of great importance is Mönting's demonstration of two results. First, he shows that M(5,42) = 22, from which it follows that the FJA can be surpassed at t = 47. This was conjectured in [1], though on weak grounds, and does not follow from Christen's method, for which 50 is the lowest value of t for which the FJA can be surpassed. (Many of the people in the field, the present author included, feel that t = 47 is the lowest value for which merging methods will work to surpass the FJA.) Second, he shows that M(m,n), m = 1, 2, 3, 4 can never be the basis for improving the FJA; in fact, he shows that

$$F(n+m) \le F(n) + F(m) + M(n,m) \quad \text{whenever } m = 1,2,3,4 \tag{4}$$

On the other hand, it can be proven easily from his formula for M(5,n) that for all $n = u_k$, $k \ge 5$,

$$F(n+5) = F(n) + F(5) + M(5,n) + 1 \tag{5}$$

We now state a number of simple lemmas easily derived from (2) and (3).

Lemma 1 [2]:

$$HL(m,n) - ITB(m,n) = D(m,n) < m \tag{6}$$

where HL(m,n) is the number of comparisons required by the HLA and ITB(m,n) is the information-theoretic lower bound $\left\lceil \log_2 \binom{m+n}{m} \right\rceil$. The function D(m,n) is studied in [2]. The relevant facts are (1) for large m,n, it is dominated by a term $m^*I(n/m)$; (2) I(a) = 0 when a = 1, and grows rather slowly toward 1 as $a \to \infty$. Some values of I(a) are shown in table 1.

a	I(a)
1	0
2	.246
4	.390
8	.471
16	.513
32	.579

Table 1

This table indicates how impressive Christen's algorithm is: For instance, for a = 4, it cuts off 64% of the distance between the HLA and the ITB.

Lemma 2 (Ford and Johnson, [12]). For all t satisfying $u_k \le t < u_{k+1}$, $F(t+1) - F(t) = k + 1$. □

From lemma 2, there follows immediately

Lemma 3. If for some t_0 in the region $u_k \le t_0 < 2^{k+1}$ (note that $2^{k+1} \approx (3/2)u_k$) it is the case that $G(t_0) < F(t_0)$, where $G(t_0)$ is the number of comparisons required by some sorting algorithm G, then for all t satisfying $t_0 \le t \le 2^{k+1}$, $G(t) < F(t)$.

Proof: Sort the t elements by first sorting t_0 elements using G, then inserting the remaining $t - t_0$ elements one at a time by binary insertion. Since binary insertion requires $k + 1$ comparisons, it follows from lemma 2 that the difference $F(t_0) - G(t_0)$ will be preserved for all t in $t_0 \le t \le 2^{k+1}$ by this process. ▯

Lemma 4. For all $k \ge 5$, $u_k + m$ elements may be sorted faster than the FJA whenever $5 \le m \le 2^{k+1}$.

Proof: By (5), this is possible for $m = 5$; by lemma 3, it is therefore possible for $5 \le m \le 2^{k+1}$. ▯

Definition 0. A region of improvement is a stretch of values of t for which improvement over the FJA is possible.

Part I. We now embark upon the main line of investigation in the first half of this paper: To discover the maximum possible extension of the region of improvement made possible by the results of Christen and Mönting. We shall first consider possible improvements using Christen's work, then we shall show that no further extension can be derived from Mönting's work, other than extending the left edge of the region via lemma 4.

Our strategy will be successively to merge sorted lists of size u_{k-2} or less into a sorted list of size u_k. We note that Christen's algorithm begins to improve over the HLA at $n = 3m$; however, we shall show later that it does no good to consider merging pieces of size n where $3m \le n < 4m$.

We first need a few more results. We note [2] that if $n = \beta m$, where $\beta = 2^{k+f}$, k is an integer, and $0 < f < 1$, then

$$HL((m,n) = m(2 + \log \beta - \theta(1 - f)) \tag{7}$$

where as before, $HL(m,n)$ is the number of comparisons required by the Hwang-Lin algorithm and $\theta(y) = 2 - y - 2^{1-y}$.

In [1], we showed that for $t = u_k + m$

$$F(u_k + m) - (F(u_k) + F(m) + HL(m,u_k)) = -\tfrac{1}{2} \log t - O(1) \tag{8}$$

for all $m < u_{k-1}$. We now consider the problem of sorting $m + n$ elements by first sorting m, then n, using the FJA, then merging using the HLA, when n is not a cusp u_k. We would expect a linear term on the right-hand side if n is not u_k for some k. Indeed, it may be calculated from (2), (3), and (7) that if $n = ru_k$ and $m = qu_{k-d}$, $1 \le r,q < 2$, and $t = n + m < u_{k+1}$, then

$$F(t) - (F(n) + F(m) + HL(m,n)) = - m\,\Delta(r,q) - O(\log t) \tag{9}$$

where

$$\left.\begin{aligned} \Delta(r,q) &= 1 - (2 - r)/q && \text{if } r \geq q \quad (10a) \\ &= (2/q)(r - 1) && \text{if } q \geq r \quad (10b) \end{aligned}\right\}$$

Of interest is the case in which $q = 1$, so that $m = u_{k-d}$. In this case, $\Delta(r,q) = r - 1$. This means that the "deficit" incurred when merging u_{k-d} elements with ru_k elements will simply be proportional to r.

Definition 1. Let us merge m elements with n. The disadvantage is defined as $m\,\Delta(r,q)$, as in (9) and (10) above. Denote the disadvantage $DA(qu_{k-d}, ru_k)$. Multiplying (10) through by m, one finds that

$$DA(qu_{k-d}, ru_k) = u_{k-d}((\min(r,q) - 1) + (r - 1)) \tag{10c}$$

Therefore disadvantage is an intrinsic function of the properties of the FJA and the HLA.

In contrast, we define advantage entirely in terms of the degree to which a new algorithm is a better merging algorithm than the HLA. We put

Definition 3: The advantage in a merge of m elements with n, $m \leq n$, is defined as $HL(m,n) - K(m,n)$, where K is the number of comparisons required by the algorithm used.

Definition 4. The total advantage (TA) of a merge is the advantage minus the disadvantage.

Note: The TA may be positive, negative, or zero, depending on the algorithm used, the sizes of the components merged, and the value of r.

Definition(s) 5. In the sequel, we assume that lists to merged have already been sorted by the FJA. Suppose our general strategy is: merge m_1 elements with ru_k elements, then m_2 elements with $(ru_k + m_1)$, then m_3 with $(ru_k + m_1 + m_2)$, etc. We will call $\sigma = (m_1, m_2, \ldots, m_j)$ a merge sequence. $\sum_i m_i$ will be called the length of the sequence and will be denoted $L(\sigma)$. In the sequel, we shall assume that $ru_k + L(\sigma) < u_{k+1}$. For $1 \leq i \leq j$, the sum $\frac{1}{u_k} \sum_{s=1}^{i-1} m_s$ will be called the effective r-value of m_i and will be denoted r_i^*, or simply r^* when i is understood. Frequently m_i will be used to denote both the i^{th} component of the merge sequence and the number of elements in that i^{th} component.

The advantage in merging m_i with ru_k will be denoted $A(m_i, ru_k)$, the disadvantage by $DA(m_i, ru_k)$ and the total advantage by $TA(m_i, ru_k)$. The accumulated total advantage for a sequence $\sigma = (m_1, \ldots, m_j)$ will be defined as $\sum_{k=1}^{j} TA(m_k, r_k^*)$ and is denoted $ATA(\sigma, ru_k)$. Accumulated advantage (AA) and accumulated disadvantage (ADA) are similarly defined. Each m_i will be called a component of σ. $ATA(\sigma, u_k)$ indicates the number of comparisons saved over $F(u_k + L(\sigma))$ by merging σ into u_k.

Lemma 5. Let $\sigma = (m_1, \ldots, m_j)$ and $L(\sigma) = u_{k-d}$. For any r, $DA(u_{k-d},ru_k) < ADA(\sigma,ru_k)$ whenever σ has more than one component. In words, if a component with length u_k is broken up into pieces, the total disadvantage increases.

Proof: From (10c), it is trivial that if m is a cusp value, then $DA(m,ru_k) = m(r - 1)$. It is also easy to show using (10c) that for *any* m, $DA(m,ru_k) \geq m(r - 1)$. Using this last formula as a lower bound, it follows that the lower bound summed over the components of σ must exceed u_{k-d} whenever σ contains more than one component. ▯

Lemma 6. If $A(m,ru_k)$ is proportional to m and independent of r, then $TA(u_{k-d},ru_k) > ATA(\sigma,ru_k)$ when $L(\sigma) = u_{k-d}$ and σ contains more than one component.

Proof: The same as for lemma 5, with the additional remark that the accumulated *advantage* for σ must be the same as the sum the advantage of its components because of the proportionality. ▯

Lemma 7. If the advantage $A(m,ru_k)$ is determined by Christen's merging algorithm ("CMA" hereafter), $TA(u_{k-d},ru_k)$ is not necessarily $\geq ATA(\sigma,ru_k)$ as in the last lemma; there is one exceptional case.

Proof: According to the CMA, the advantage for n/m in the range (2,4) is m/4; for n/m in the range (4,6) is 5/16; for n/m in the range (6,8) is 21/64, etc.

Let us divide up u_{k-d} into pieces that are all cusp values, i.e., $u_{k-d'}$, $u_{k-d''}$, etc. This process can be viewed as a sequence of divisions in half of various components.

If a component is divided in half, it is not hard to show that the increase in the ADA will go like $m/2^{d+2}$. On the other hand, there may or may not be a gain in the AA; if d is odd there will be, but if d is even there will not be. From Christen's formula given earlier, the gain for odd d will be $m/2^d$. So there will be a net gain in the ATA if d is odd and a net loss if d is even. The process of dividing any component of size u_{k-d} into pieces of assorted sizes $u_{k-d'}$, $u_{k-d''}$, etc., will, however, *not* in general produce an increase in ATA. Only the single division into halves when d is odd will produce an increase. ▯

Definition 6. Given two sequences σ_1 and σ_2, with $L(\sigma_1) = L(\sigma_2)$, σ_1 will be called *better than* σ_2 *at* r iff $ATA(\sigma_1,ru_k) > ATA(\sigma_2,ru_k)$.

Definition 7. An *advantage graph* is a function G(r), with ordinate between u_k and u_{k+1}. The function is piecewise linear and positive; the i^{th} piece p_i denotes m_i in σ. The abcissa of the right end of p_i displays $ATA((m_1, \ldots, m_i),u_k)$. A *node* or *vertex* is said to occur at the right end of each p_i.

Corollary 7.1. The slope of p_i is positive, zero, or negative, according as $TA(m_i)$ is likewise.

Definition 8. A sequence σ will be called optimal in the interval (r_1, r_2) if (a) $L(\sigma) = (r_2 - r_1)u_k$ and (b) $ATA(\sigma, r_1 u_k) \geq ATA(\sigma', r_1 u_k)$ for all other sequences σ' of length $L(\sigma)$. A sequence will be called optimal at r if it is optimal in $(1, r)$.

Lemma 9. Using the CMA, either the sequence (u_{k-d}) or the sequence (u_{k-d-1}, u_{k-d-1}) is optimal in any region (r_1, r_2) for which $r_2 - r_1 = u_{k-d}$.

Proof: Follows from lemma 7 and the discussion in its proof. []

From here on, we shall always assume that advantage is produced by the CMA.

Lemma 10. An optimal sequence in the interval (1,3/2) has the property that it can be composed of two parts:
(a) The sequence σ_1 with the largest possible value of $ATA(\sigma,1)$ in the region $(1,r_0)$ for all r_0 satisfying $1 \leq r_0 \leq 3/2$;
(b) A sequence of ones of length $3/2 - L(\sigma_1)$.

Proof. From the proof of lemma 3, it is clear that the ATA for σ_1 can be carried across to $r = 3/2$ by the simple expedient of inserting elements one at a time. From this fact, one needs only find that σ_1 having the largest possible ATA in $(1,r_0)$. []

Definition 8a. A sequence σ will be said to have best reach if $ATA(\sigma,u_k) \geq 0$ for σ of greatest possible length; that is, for any other σ' with $L(\sigma') > L(\sigma)$, $ATA(\sigma',u_k) \leq 0$. The quantity $u_k + L(\sigma)$, where σ has best reach, will be called the maximum reach.

Observe that the maximum reach is a function purely of the advantage function, and therefore of the merging algorithm. Denote the maximum reach, given a merging algorithm α, by $MR(\alpha)$.

Lemma 11. $MR(CMA) \geq 7/4$.

Proof: Consider the simple merge sequence $(u_{k-2}, u_{k-2}, u_{k-2})$. The advantage for each component is m/4. The disadvantages are resp. (0, m/4, m/2). This immediately yields the advantage graph of figure 1. []

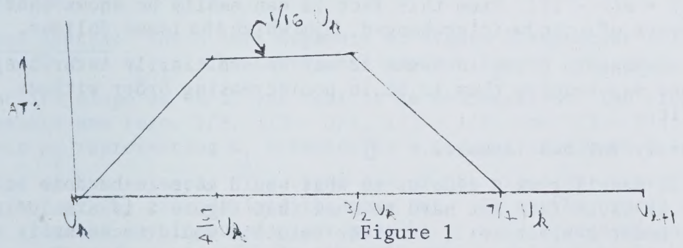


Figure 1

Conjecture 1. A merge sequence of best reach for the CMA is achievable in which all the components are cusp values, i.e., for all i, $m_i = u_{k-d_i}$ for $d_i \geq 2$.

Lemma 12. There exists a σ with positive ATA obeying conjecture 1 and $L(\sigma) > .75$, in particular $MR(CMA) \geq 1.76$.

Proof: The following construction.

Let $\sigma = \sigma_1 \sigma_2 \sigma_3 \sigma_4$, where
$\sigma_1 = (u_{k-2}, u_{k-4}, u_{k-6}, \ldots)$, so that $L(\sigma_1) = 1/3\ u_k$;
$\sigma_2 = 1$ repeated $1/6\ u_k$ times;
$\sigma_3 = (u_{k-2})$;
$\sigma_4 = (u_{k-d_1}, u_{k-d_2}, \ldots)$ where d_i are rather large numbers chosen so that $L(\sigma_4) \simeq 1/100$.

The construction works in this way. u_{k-2} is first merged. At this point, $r = 5/4$, so that merging u_{k-2} again would produce zero TA. However, by the CMA, a component of size u_{k-4} will have advantage 5/16, therefore total advantage $u_{k-4}(5/16 - (5/4 - 1))$ which is positive. In this way, we obtain $L(\sigma_1) = 1/3$ and $ATA(\sigma_1, u_k) = 1/15\ u_k$. From lemma 3, we know that $L(\sigma_1 \sigma_2) = 1/2\ u_k$ and $ATA(\sigma_1 \sigma_2, u_k) = 1/15\ u_k$. It is easy to show that $ATA(\sigma_1 \sigma_2 \sigma_3, u_k) = (1/15 - 1/16)\ u_k = 1/240\ u_k$. The last pieces of σ_4 are very small, so that the advantage for each component c of it is about $c(1/3 - 3/4) = -5/12\ c$. Therefore $L(\sigma_4) \simeq 12/5*240 = 1/100$, and $L(\sigma) \simeq .760$. ▯

The graph of this construction is shown in figure 2.

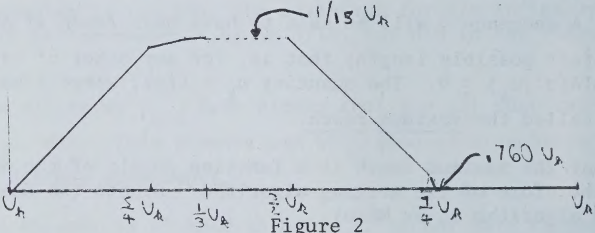


Figure 2

Lemma 13. If $\sigma = (m_1, \ldots, m_j)$, and all m_i are cusp values, then using the CMA, $ATA(\sigma, u_k) = ATA(\sigma', u_k)$, where σ' is any permutation of σ.

Proof. The advantages under the CMA are additive. By (10c), for m a cusp value, $DA(m, ru_k) = m(r - 1)$. From this fact is can easily be shown that two adjacent components of σ can be interchanged, from which the lemma follows. ▯

Lemma 14. The components of σ_1 in lemma 12 may be arbitrarily interchanged; in particular one may require them to be in nonincreasing order without loss of generality.

Proof: Immediately follows lemma 13. ▯

Lemmas 3, 7, and 14 give a clue as to what would have to be done to beat figure 2. It is in fact not hard to show that figure 2 is a solution with best reach under conjecture 1; a better solution would necessarily entail non-cusp size components. Therefore we may assert

Conjecture 2. Figure 2 is a graph for a sequence of maximum reach given the CMA, and $MR(CMA) = 1.76$.

We now show that the CMA cannot possibly beat the FJA everywhere. In particular, we now show that MR(CMA) is less than 2.

Definition 9. The hull of a set of graphs (cf. definition 7) is a curve $f(r)$, such for a given merging method and any advantage graph G,

$G(r) \le f(r)$ for all r.

In other words, $f(r)$ dominates all advantage graphs from above.

Lemma 15. The curve of figure 3 is a hull of all graphs over the CMA.

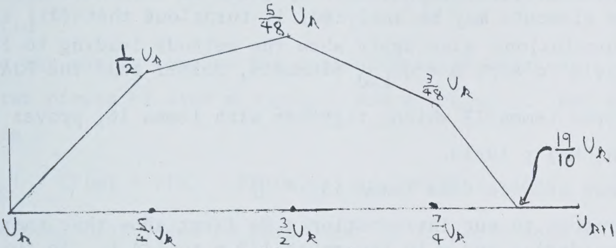


Figure 3

To prove this, we first need a few more definitions and lemmas.

Definition 10. Consider a merge sequence σ and a component m. We say that we attach m to σ by forming a new sequence σm.

Definition 11. Given $\sigma = \sigma_1 \sigma_2$, with $L(\sigma_1) = x$, we say that we attach m at x by forming $\sigma_1 m$ and discarding σ_2.

Definition 12. Given some graph H, if we attach some m to H at x and then find that $H(x + m) < H(x) + TA(m)$, we will say that x sticks out of H.

Lemma 16: If no permissible component (i.e., one whose TA is given by (10c) and the CMA) attached to H at any x sticks out of H, then H is a hull.

Proof: If it is not a hull, then some advantage graph G must penetrate it somewhere. Now H is convex downward, so some vertex of G must penetrate. Suppose m^* is the component whose right end is this (penetrating) vertex. Denote the right end of m^* by R and the left end by L. If $G(L) > H(L)$, then H is not a hull. Therefore assume $G(L) \le H(L)$, in this case, attaching m^* to H at the same ordinate position where it attaches to G raises it by a constant, so that it sticks out "even further." This, by negation, proves lemma 16. ▯

Lemma 17. No component permitted by the CMA, when attached to H, penetrates H.

Proof Sketch: The 3 left segments of figure 3 represent 3 inserted components of size u_{k-2}. The fourth has maximum length subject to the constraint that its slope be -5/12 and that it be nonnegative. The slopes of the 4 segments are resp. 1/3, 1/3 - 1/4, 1/3 - 1/2, and 1/3 - 3/4. Therefore, a piece p_i representing m_i attached to a segment of H must have slope $\le$ the slope of the segment. The right end of the piece is either (a) in the same region, or (b) extends into the next one. It may be proven from (10) that the slope of any inserted piece must be $\le 1/3 - x$ regardless of the value of q. Therefore case (b) is the only one in which H might be penetrated.

Consider the case in which $m = u_{k-d}$, $d \ge 2$, and m attaches to H in the first region. Let m attach at x_a, $x_a < 1/4$. Let $A(m) = 1/3\ m$. Then the requirement that the hull be above the right end of the attached component can be expressed as

$$1/3\ x_a u_k + m(1/3 - x_a) \le 1/12\ u_k + 1/12(m + x_a u_k - 1/4\ u_k) \qquad (11)$$

which is an equality for $m = u_{k-2}$ and is satisfied for $m = u_{k-d}$, $k \geq 3$, since $x_a \geq 1/4 - m$.

(11) can be shown to hold in the other regions, and also to hold when $q > 1$ because of our remark on slopes. Finally, if $m = qu_{k-d}$, $d \geq 3$, the possibility of additional improvement by means of recursive application of figure 2 to m elements may be analyzed; it turns out that (11) still holds.

These conclusions also apply when the methods leading to figure 2 are used recursively to sort $m = qu_{k-d}$ elements, rather than the FJA.

This proves lemma 17 which, together with lemma 16, proves lemma 15. []

Lemma 18. MR(CMA) $\leq$ 19/10.

Proof: Follows at once from lemma 15. []

We now return to our introduction. We first show that there is never profit in considering any m in the range 1/3 n to 1/4 n. In this region, one can show at once from [5] that the advantage goes like $m(1/4(n/m) - 3/4)$, so that the advantage declines as the length increases. It is possible to show for m in this range that one is always better off breaking m into two pieces, one of length u_{k-2} and the other of length $m - u_{k-2}$. The second point raised in the introduction is handled by the next lemma.

Lemma 19. The improvements due to Mönting [7] cannot affect figures 1 or 2.

Proof: Mönting's original formula for $M(5,n)$, which is either accurate or an extremely sharp upper bound, yields values of s for which $M(5,su_k)$ increases by 1, for k sufficiently large. To be more precise: As k increases, s approaches these values, rapidly and asymptotically. The smallest s-value is 1.068, and predicts that for u_k to $1.068u_k$, each 5 elements inserted will better the FJA by one comparison. Iterating this process corresponds to an advantage of 1/5 m in this region, whereas Christen's algorithm gives an advantage of 1/4 m. Next, we might hope that the flat region in figure 2 from 1/3 to 1/2 could be improved. Alas! The flat region (zero advantage) for inserting 5 elements runs from s = 1.068 to s = 1.242. The region in which the advantage goes to -1/5 m runs from s = 1.242 to 1.429. Note that these are total advantages; Mönting's method does not obey (10c)! Finally, the effective TA is -2/5 from s = 1.429 to 1.627. This is of no help, because Christen's method gives a TA of -1/4 at s = 1/2, which we then use to insert a large component.

Thus, Mönting's method helps only to establish the lower bound of lemma 4 but not elsewhere. []

Part II. Having established that there exists at the present time no known merging method that can improve on the FJA for all values of r, in partilar, for $19/10 < r \leq 2$, let us now ask whether any hypothetical merging method, even one given only as a lower bound, might suffice.

The most obvious thing to do is to ask the simple question: If merging could be accomplished in the number of comparisons given by the information-theoretic lower bound of

$$\mathrm{ITB}(m,n) = \left\lceil \log_2 \binom{m+n}{m} \right\rceil$$

could we sort t elements by partitioning t judiciously, use the algorithm recursively to sort the pieces, then recombine them in some judicious order using the ITB? Would this process beat the FJA at its cusp values?

From Stirling's approximation we may calculate

$$F(ru_k) - (2F(ru_{k-1}) + ITB(ru_{k-1},ru_{k-1})) = O(1) \qquad (12)$$

The O(1) term turns out to be small and nonpositive, therefore we have

Lemma 20. If the FJA can be beaten at $t = ru_k$, then it can be beaten at all $t' = ru_{k+d}$ for d = 1, 2, ... □

We also can easily compute the consequences of breaking $t = u_k$ elements into two pieces of size $m = q_1 u_{k-d}$ and $n = q_2 u_{k-1}$, Ket $q_1 = 2^f$ and $q_2 = 2^g$, then

$$F(u_k) - (F(m) + F(n) - ITB(m,n)) = -(m\theta(f) + n\theta(g)) - O(1) \qquad (13)$$

With θ as before, (13) seems to imply that it is profitless to break up u_k into two pieces. But (13), of course, assumes that the sets m and n were sorted by the FJA, whereas we are not making that assumption.

Let E(t) denote the minimum number of comparisons to sort t elements, given (a) that t is partitioned arbitrarily, (b) that the partitioned pieces are sorted using E recursively, or by the FJA, whichever is best, and (c) that the merging is done in some "best" order. A simple induction shows that E(t) is computed equally well by changing (b) to read "that the partitioned pieces are sorted using the FJA."

This much having been done, it is not hard to see that E(t) can be computed by dynamic programming using the following inductive idea: The last merge in step (c) must be a merge of two pieces m and n for which resp. E(m) and E(n) comparisons are required. Therefore.

$$E(t) = \min(D(t),F(t)), \text{ where}$$

$$D(t) = \min_{1 \le p < t} (E(p) + E(t-p) + ITB(p, t-p)) \qquad (14)$$

We have run a computer program to compute E(t). In general, E(t) is less than F(t) except in the vicinity of u_k for small k. In these regions, a few consecutive values of t yield E(t) = F(t), but the number of such values decreases with increasing k, finally disappearing altogether for $t = u_{11} = 2730$. This fact, together with (12) proves

Lemma 21. If information-theoretic merging were achievable, then there would exist a constant t_0 such that for all $t > t_0$, the FJA could be surpassed. Further, $t_0 \le u_{10} = 1365$. □

Hypothesizing that the "real" bound is one larger than ITB(m,n) for $m \ge 5$ yields the same result as lemma 21, except that $t_0 \le u_{11}$.

We therefore searched for some nontrivial, small, yet plausible function that we might add to the ITB in order to cause the number of values of t surrounding each u_k for which E(t) = F(t) to increase. No linear improvement (proportional to m) is known. However, it is surmised that for n/m larger than some constant, it may well be that the real lower bound is the ITB plus a small term proportional to m. This would still not be enough.

We noticed, however, that empirically the following formula appears to be true:

$$M(n,m) \quad ITB(m,n) + \lfloor \tfrac{1}{2} \log_2 (m-1) \rfloor \quad \text{where } m \geq 5 \tag{15}$$

where as usual $m \leq n$. Plugging this formula into the dynamic program together with Mönting's values of $M(m,n)$ for $m = 1, 2, 3, 4$, and 5 produces a slightly growing number of values of t surrounding each u_k for which $E(t) = F(t)$, at least as far as we have taken it. This number, $N(k)$, is tabulated in table 2.

k	u_k	$N(k)$
5	42	14
6	85	16
7	170	21
8	341	26
9	682	30
10	1365	37
11	2730	41

Table 2

The results suggest that perhaps the following conjecture may be true.

Conjecture 3. There exists no method for partitioning t elements, then merging the pieces, that will beat the FJA for any $t = u_k$.

REFERENCES

1. Manacher, G.K. "The Ford-Johnson Algorithm is not optimal." J. of the ACM, 26, 3, July 1979, pp. 441-456.
2. Knuth, Donald. The Art of Computer Programming. Vol. 3, Addison-Wesley, 1973, pp. 181-209.
3. Hwang, F. and Lin, S. "A Simple Algorithm for Merging two Disjoint Linearly-Ordered Sets." SIAM J. Comp., 1, 1, 1972, pp. 31-39.
4. Manacher, G.K. "Significant Improvements to the Hwang-Lin Merging Algorithm." J. of the ACM, 26, 3, 1979, 434-440.
5. Christen, C. Improving the Bounds on Optimal Merging. Proc. of the 19th Annual FOCS Conference, Ann Arbor, Michigan, 1978, 259-266.
6. Christen, C. Private Communication, August 1978.
7. Mönting, Schulte. J. Optimal Merging of 3, 4, and 5 elements with N elements, Technical Report, 1978, Mathematisches Institut, Tübingen, Germany.
8. Graham, R.L. On Sorting by Comparisons. Proc. of the Second Atlas Conference, 1971, 263-269.
9. Hwang, F. and Lin, S. "Some Optimality Results in Merging two Elements with n Elements." Acta Informatica 1, 1, 1971, 145-158.
10. Hwang, F.K. Manuscript Tech. Report, Bell Laboratories, 1977.
11. Nissenbaum, Y. Reference not available at time of writing.
12. Ford, L.R., Jr. and Johnson, S.B. "A Tournament Sorting Problem." AMM 66, 5, 1959, 387-389.

THE UP-DOWN ALGORITHM FOR TWO-DIMENSIONAL BIN PACKING

BRENDA S. BAKER
Bell Laboratories
Murray Hill, NJ 07974

DONNA J. BROWN*
Coordinated Science Laboratory
University of Illinois
Urbana, IL 61801

HOWARD P. KATSEFF
Bell Laboratories
Holmdel, NJ 07733

SUMMARY

The two-dimensional bin packing problem, first proposed in [1], is as follows: a finite list L of rectangles is to be packed into a rectangular bin of finite width but infinite height, so as to minimize the maximum height used. The packed rectangles cannot overlap, nor can they be rotated. Since the problem of finding an optimal packing is NP-hard, we investigate approximation algorithms. Let OPT(L) denote the minimum possible bin height for a packing of L and A(L) the height used by a particular algorithm A when applied to list L. For an algorithm A, we study the asymptotic performance bound,

$$A(L) \leq \alpha \cdot OPT(L) + \beta,$$

for all lists L (α,β are constants). The best algorithm previously known, the Mixed algorithm [3], has $\alpha = 4/3$. We propose a new algorithm, the Up-Down (UD) algorithm, which has an asymptotic bound of 5/4.

The UD algorithm divides the rectangles of list L into five sublists L_i $(1 \leq i \leq 5)$ and successively packs L_1, L_2, L_3, L_4, L_5 into regions R_i. Assume that the bin width is normalized to 1. For $1 \leq i \leq 4$, L_i consists of pieces with widths in the range $(\frac{1}{i+1}, \frac{1}{i}]$, ordered by nonincreasing width. L_5 is a list of the pieces of L with width at most $\frac{1}{5}$, such that the pieces are in order of nonincreasing height. Within each region R_i $(1 \leq i \leq 4)$, pieces of L_i are packed from the bottom left using the BL algorithm [1]. Except for small gaps between the BL pieces, this leaves any unfilled space toward the top and right of R_i. So along the right edge of the bin within each region R_i, a column is packed top down using pieces of L_j $(i+1 \leq j \leq 4)$. Finally, pieces of L_5 are packed by NFDH [2], wherever possible between the BL pieces and the right column pieces of R_i $(1 \leq i \leq 4)$, otherwise in R_5.

[1] B. S. Baker, E. G. Coffman, and R. L. Rivest, "Orthogonal Packings in Two Dimensions," SIAM Journal of Computing, to appear.

[2] E. G. Coffman, M. R. Garey, D. S. Johnson, and R. E. Tarjan, "Performance Bounds for Level-Oriented Two-Dimensional Packing Algorithms," submitted for publication (1979).

[3] I. Golan, "Performance Bounds for Orthogonal Oriented Two-Dimensional Packing Algorithms," draft (1979).

---

* This work was supported by the Joint Services Electronics Program (U.S. Army, U.S. Navy, and U. S. Air Force) under Contract DAAG-29-78-C-0016.

AN ALGORITHM FOR THE HITCHCOCK TRANSPORTATION PROBLEM

DIMITRI P. BERTSEKAS
Dept. of Electrical Engineering and Computer Science
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology

Abstract: An algorithm proposed by the author for the classical assignment problem [1] is generalized for solution of the linear uncapacitated transportation problem.

1. INTRODUCTION

In an earlier paper [1] we proposed a new algorithm for solving the classical assignment problem. The algorithm was shown via computational experimentation to offer substantial computational savings over the Hungarian method. It is thus natural to consider extensions of this algorithm to more general network flow problems. The present paper provides such an extension for the classical linear uncapacitated transportation problem, commonly referred to as the Hitchcock problem. As is well known [2], the general minimum cost flow problem with capacity constraints on the links can be reduced to the Hitchcock problem. Thus the algorithm of this paper can be adapted to solve such problems although we will not discuss the precise form of the necessary modifications.

Due to space limitations our presentation is somewhat abbreviated. It is thus inevitable that some familiarity with the contents of [1] on the part of the reader is necessary for understanding the mechanism of the algorithm.

2. THE HITCHCOCK PROBLEM

Consider a bipartite graph consisting of two finite sets of nodes S and T with elements denoted i (i=1,...,M) and j (j=1,...,N) respectively, and a set of directed links L with elements denoted (i,j) where $i \in S$ and $j \in T$. We refer to elements of S and T as sources and sinks respectively. Each source i (sink j) has a positive scalar α_i (β_j) associated with it referred to as the supply of i (demand of j). We assume

$$\sum_{i \in S} \alpha_i = \sum_{j \in T} \beta_j . \tag{1}$$

Each link (i,j) has a weight a_{ij} associated with it. We assume that there is at most one link $(i,j) \in L$ for all $i \in S$, $j \in T$. We wish to find a flow $x = \{x_{ij} \mid (i,j) \in L\}$ solving the (primal) transportation problem

$$\text{maximize} \quad \sum_{(i,j) \in L} a_{ij} x_{ij}$$

$$\text{subject to} \quad \sum_{(i,j) \in L} x_{ij} = \alpha_i , \qquad \forall\, i \in S$$

$$\sum_{(i,j) \in L} x_{ij} = \beta_j , \qquad \forall\, j \in T$$

$$x_{ij} \geq 0 , \qquad \forall\, (i,j) \in L .$$

Throughout the paper we will assume the following:

a) There exists at least one feasible solution for (PTP).

b) The weights a_{ij}, the supplies α_i, and the demands β_j are all integers.

The corresponding dual transportation problem in the vectors $m = (m_1,\ldots,m_M)$ and $p = (p_1,\ldots,p_N)$ is

$$\text{(DTP)} \quad \text{minimize} \sum_{i\in S} \alpha_i m_i + \sum_{j\in T} \beta_j p_j$$

$$\text{subject to } m_i + p_j \geq \alpha_{ij} \quad , \qquad \forall\, (i,j)\in L \,.$$

The scalars p_j and $(a_{ij} - p_j)$ will be referred to as prices and profit margins respectively.

3. The Algorithm

The algorithm generates a sequence of vectors $\{(m^k, p^k, x^k)\}$ such that for each k the following conditions are satisfied:

$$m_i^k + p_j^k \geq a_{ij} \quad , \qquad \forall\, (i,j)\in L, \tag{2}$$

$$m_i^k + p_j^k = a_{ij} \quad , \qquad \forall\, (i,j)\in L \text{ with } x_{ij}^k > 0 \,, \tag{3}$$

$$x_{ij}^k \geq 0 \quad , \qquad \forall\, (i,j)\in L \tag{4}$$

$$\sum_{(i,j)\in L} x_{ij}^k \leq \alpha_i \quad , \qquad \forall\, i\in S, \tag{5}$$

$$\sum_{(i,j)\in L} x_{ij}^k \leq \beta_j \quad , \qquad \forall\, j\in T \quad . \tag{6}$$

Thus dual feasibility and complementary slackness are maintained throughout the algorithm [cf.(2),(3)], but primal feasibility may be violated to the extent allowed by (4) - (6). The algorithm, under the preceding assumptions, can be shown to terminate at a flow for which (5) and (6) are satisfied with equality. Since in this case primal and dual feasibility as well as complementary slackness are satisfied, this flow must be optimal.

The overall scheme is similar with the one of the primal-dual method ([2], p.95), in that successive flow augmentations and changes in the dual variables are effected, but there are important differences as discussed in [1]. We now state the algorithm.

The method is initialized with any m^o, p^o that are dual feasible, and with $x_{ij}^o = 0$ for all $(i,j)\in L$.

For k = 0,1,..., given (m^k, p^k, x^k) satisfying (2) - (6) we stop if (5) and (6) are satisfied with equality. Otherwise we use the following iteration to generate $(m^{k+1}, p^{k+1}, x^{k+1})$. During the iteration additional intermediate variables, vectors and sets of nodes are generated, and are denoted by $\overline{m}$, $\tilde{m}$, $\overline{T}$, $\tilde{T}$, $\overline{x}, \pi, f$.

(k+1)st Iteration of the Algorithm

Choose a source $\overline{i}$ such that $\displaystyle\sum_{(i,j)\in L} x_{\overline{i}j}^k < \alpha_{\overline{i}}$

Set $\overline{m}$ to the maximum profit margin for $\overline{i}$

$$\bar{m} \leftarrow \max\{a_{\bar{i}j} - p_j^k \mid (\bar{i},j) \in L\}^{\dagger} \quad . \tag{7}$$

Set also

$$\bar{T} \leftarrow \{j \in T \mid \bar{m} = a_{\bar{i}j} - p_j^k\} \quad , \tag{8}$$

$$\tilde{T} \leftarrow \bar{T} \quad . \tag{9}$$

Step 1 (Preliminary adjustment of flows and dual variables): If

$\alpha_{\bar{i}} < \sum_{j \in \bar{T}} \beta_j$ go to Step 2. If $\alpha_{\bar{i}} \geq \sum_{j \in \bar{T}} \beta_j$ go to Step 1a.

Step 1a: Set

$$\tilde{m} \leftarrow \max\{a_{\bar{i}j} - p_j^k \mid (\bar{i},j) \in L, \ j \notin \bar{T}\}. \tag{10}$$

$$\tilde{T} \leftarrow \{j \in T \mid \tilde{m} = a_{\bar{i}j} - p_j^k\} \tag{11}$$

$$p_j^k \leftarrow \begin{cases} p_j^k & \text{if } j \notin \bar{T} \\ p_j^k + (\bar{m} - \tilde{m}) & \text{if } j \in \bar{T} \end{cases} \tag{12}$$

$$m_i^k \leftarrow \begin{cases} m_i^k & \text{if } i \neq \bar{i} \\ \tilde{m} & \text{if } i = \bar{i} \end{cases} \tag{13}$$

$$x_{ij}^k \leftarrow \begin{cases} x_{ij}^k & \text{if } j \notin \bar{T} \\ \beta_j & \text{if } i = \bar{i} \text{ and } j \in \bar{T} \\ 0 & \text{if } i \neq \bar{i} \text{ and } j \in \bar{T} \ , \end{cases} \tag{14}$$

If $a_{\bar{i}} = \sum_{j \in \bar{T}} \beta_j$ set

$$m^{k+1} \leftarrow m^k$$

$$p^{k+1} \leftarrow p^k$$

$$x^{k+1} \leftarrow x^k$$

thereby completing the (k+1)st iteration.

If $\alpha_{\bar{i}} \neq \sum_{j \in \bar{T}} \beta_j$ set

$$\bar{T} \leftarrow \bar{T} \cup \tilde{T}$$

$$\bar{m} \leftarrow \tilde{m}$$

and go to Step 1.

[†] The notation A ← B means that the current value of (the variable, vector, or set) A is changed to the current value of B.

Step 2.

(Note: At this point we have from Step 1 vectors m^k, p^k, x^k and nonempty sets of sinks $\bar{T}$, $\tilde{T}$ with $\tilde{T} \subset \bar{T}$ such that (2) - (6) hold and

$$\sum_{j\in(\bar{T}-\tilde{T})} \beta_j = \sum_{j\in(\bar{T}-\tilde{T})} x^k_{\bar{i}j} \le \sum_{(\bar{i},j)\in L} x^k_{\bar{i}j} < \alpha_{\bar{i}} < \sum_{j\in\bar{T}} \beta_j \quad , \tag{16}$$

$$x_{\bar{i}j} = \beta_j \, , \qquad \forall\, j\in(\bar{T}-\tilde{T}), \tag{17}$$

$$m^k_{\bar{i}} + p^k_j = a_{\bar{i}j} \, , \quad \forall\, j\in\bar{T} \quad , \tag{18}$$

where if $\bar{T} = \tilde{T}$ we interpret sums over $j\in(\bar{T}-\tilde{T})$ to be zero).

If there is no unassigned demand for sinks in $\tilde{T}$, i.e. if

$$\sum_{j\in\tilde{T}} \beta_j = \sum_{j\in\tilde{T}} \sum_{(i,j)\in L} x^k_{ij} \quad ,$$

go to Step 3.

Otherwise increase the flow $x^k_{\bar{i}j}$ along links $(\bar{i},j)$ for sources $j\in\tilde{T}$ for which a portion of their demand is unassigned, (i.e., $\beta_j > \sum_{(i,j)\in L} x^k_{\bar{i}j}$) until either

a) the unassigned portion of the supply of $\bar{i}$ (i.e. $\alpha_{\bar{i}} - \sum_{(\bar{i},j)\in L} x^k_{\bar{i}j}$) is exhausted,

or

b) The unassigned portion of the demand of sinks in $\tilde{T}$ (i.e. $\sum_{j\in\tilde{T}} \beta_j - \sum_{j\in\tilde{T}} \sum_{(i,j)\in L} x^k_{ij}$) is exhausted.

In case a) set $m^{k+1} \leftarrow m^k$, $p^{k+1} \leftarrow p^k$, and set x^{k+1} to be the new flow. This completes the (k+1)st iteration.

In case b) increase further flows $x^k_{\bar{i}j}$ with $j\in\tilde{T}$ and decrease flows x^k_{ij} with $i \neq \bar{i}$, $j\in\tilde{T}$ as necessary until all unassigned supply of $\bar{i}$ is exhausted. (Note: The choice of flows $x^k_{\bar{i}j}$ which are increased and flows x^k_{ij}, $i\neq\bar{i}$, $j\in\tilde{T}$, which are decreased is arbitrary).

Set x^{k+1} to be the new flow, and set $m^{k+1} \leftarrow m^k$, $p^{k+1} \leftarrow p^k$. This completes the (k+1)st iteration.

Step 3: Increase flows $x^k_{\bar{i}j}$ with $j\in\tilde{T}$ and decrease flows x^k_{ij} with $i \neq \bar{i}$, $j\in\tilde{T}$ as necessary until all unassigned supply of $\bar{i}$ is exhausted. We denote the new set of flows $\bar{x}_{ij}$, $(i,j)\in L$ and refer to $(\bar{x}_{\bar{i}j} - x^k_{\bar{i}j})$, $j\in\tilde{T}$ as the incremental flows and to $(x^k_{ij} - \bar{x}_{ij})$, $j\in\tilde{T}$, $i \neq \bar{i}$ as the displaced flows. Give to $\bar{i}$ the label "$(0,f_{\bar{i}})$" where $f_{\bar{i}} = \alpha_{\bar{i}} - \sum_{(\bar{i},j)\in L} x^k_{\bar{i}j}$ set

$$\pi_j \leftarrow \infty \quad , \ \forall\, j\in T$$

and go to Step 4.

Step 4 (Labeling): Find a source i with an unscanned label and go to Step 4a, or find a sink j with an unscanned label and $\pi_j = 0$, and go to Step 4b. If no such source or sink can be found go to Step 6.

Step 4a: Scan the label of source i as follows. For each $(i,j) \in L$ for which $m_i^k + p_j^k - \alpha_{ij} < \pi_j$ set $\pi_j = m_i^k + p_j^k - \alpha_{ij}$. If in addition $m_i^k + p_j^k = a_{ij}$ give node j the label "(i,f_j)" where $f_j = \min\{f_i, \beta_j - \bar{x}_{\bar{i}j}\}$. Return to Step 4.

Step 4b: Scan the label of sink j with $\pi_j = 0$ as follows. If $\sum_{(i,j)\in L} \bar{x}_{ij} < \beta_j$ go to Step 5. Otherwise give to every unlabeled source i with $x_{ij} > 0$ the label "(j,f_i)" where $f_i = \min\{f_j, \alpha_i\}$. Return to Step 4.

Step 5 (Augmentation) : An augmenting path has been found that alternates between sources and sinks, originates at source $\bar{i}$ and terminates at the sink j identified in Step 4b. The flow on this path is

$$f = \min\{f_j, \beta_j - \sum_{(i,j)\in L} \bar{x}_{ij}\}$$

. Modify $\bar{x}$ by adding or subtracting as appropriate f along the augmenting path. Subtract from $\bar{x}$ a portion of incremental flows totalling f and add to x the corresponding displaced flows totalling f. The end result is the new set of flows x^{k+1}. Set $m^{k+1} \leftarrow m^k$, $p^{k+1} \leftarrow p^k$. This completes the (k+1)st iteration.

Step 6 (Change of dual variables): Find

$$\delta = \min\{\pi_j | j \in T, \ \pi_j > 0\}.$$

Set

$$m_i^{k+1} \longleftarrow \begin{cases} m_i^k - \delta & \text{if i has been labelled} \\ m_i^k & \text{if i has not been labelled,} \end{cases}$$

$$p_j^{k+1} \longleftarrow \begin{cases} p_j^k + \delta & \text{if } \pi_j = 0 \\ p_j^k & \text{if } \pi_j > 0 \text{ ,} \end{cases}$$

$$x^{k+1} \leftarrow \bar{x} \quad .$$

This completes the (k+1)st iteration of the algorithm.

The description of the algorithm is quite complicated so we provide some explanatory remarks. The objective of each iteration originating with source $\bar{i}$ is to assign all unassigned supply of $\bar{i}$ while in the process to either change some of the dual variables or assign some of the unassigned demand of certain sinks or both. This is done at the expense, perhaps, of displacing some already assigned supply of other sources. Step 1 is the preparatory stage whereby by modifying flows and dual variables if necessary [cf. (12) - (14)] we identify two sets of sinks T and $\tilde{T}$ such that (16) - (18) are satisfied. The iteration may end in Step 1 or in Step 2 after essentially scanning only source $\bar{i}$. This is the simpler of two possible cases and corresponds to Case 1 of the algorithm of [1]. The more complicated case, corresponding to Case 2 of the algorithm of [1], is when the demand of all sinks in $\tilde{T}$ are fully satisfied and therefore in order to assign even a portion of the remaining unassigned supply $\alpha_{\bar{i}} - \sum_{(\bar{i},j)\in L} x^k_{\bar{i}j}$ of source $\bar{i}$ it is necessary to displace some portion of already assigned supply of other sources. This is done by means of the labeling procedure starting with Step 3. A preliminary assignment of the unsatisfied supply of $\bar{i}$ is made by displacing assigned supply of other sources ($\bar{x}$ replacing x^k in Step 3). We then try to find an augmenting path from $\bar{i}$ to a sink with unsatisfied demand. There are two possibilities. Either such an augmenting path can be found (Step 5) in which case a portion f of the unsatisfied supply of $\bar{i}$ is chanelled through the augmenting path and the corresponding portion f of the displaced supply of other sources is reinstated; or else a change is effected in the dual variables (Step 6). The amount of change is the maximum allowed by the complementary slackness constraint.

Observe that the following hold true similarly as in the algorithm of [1]:

a) The sequences $\{m^k_i\}$ are monotonically nonincreasing and the sequences $\{p^k_j\}$ are monotonically nondecreasing.

b) The sequences of unassigned sink demands $\{\beta_j - \sum_{(i,j)\in L} x^k_{ij}\}$ are monotonically nonincreasing.

c) Dual feasibility and complementary slackness are maintained by the algorithm.

d) At every iteration either the unassigned demand of at least one sink will decrease strictly by an integer amount, or at least one price will increase strictly by an interger amount.

e) In order for the price of a sink to increase it is necessary that its demand is fully assigned.

Based on these observations we can show in a very similar manner as in [1] that the algorithm will terminate at a feasible flow. In view of the fact that complementary slackness and dual feasibility are maintained throughout the algorithm, the flow obtained at termination is optimal.

We have no computational experience with the algorithm with general transportation problems. We except, however, that similarly as for assignment problems [1], some combination with the primal-dual method will turn out to be beneficial. It is hoped that computational experience will be gained in the near future in this regard.

References

[1] D.P. Bertsekas, "A New Algorithm for the Assignment Problem", Lab. for Information and Decision Systems Working Paper, Massachusetts Institute of Technology, July 1979.

[2] L.R. Ford and D.R. Fulkerson, Flows in Networks, Princeton University Press, 1962.

Acknowledgement: This work was supported by Grant NSF ENG-7906332.

ON THE COMPLEXITY OF STEINER-LIKE PROBLEMS

P.M. CAMERINI (°), G. GALBIATI (+) & F. MAFFIOLI (°)

(°) Istituto di Elettrotecnica ed Elettronica & Centro Studi per le Telecomunicazioni Spaziali of CNR, Politecnico di Milano, 20133 MILANO (Italy)

(+) Istituto di Matematica, Università di Pavia, 27100 PAVIA(Italy)

Abstract

In this paper we investigate the computational complexity of Steiner-like problems, i.e. problems asking for the existence in a graph of a tree with bounded cost and constrained set of vertices. A borderline between easy and NP-complete problems is identified. Two kinds of Steiner-like problems are studied, namely labelled problems in which the identity of the vertices of the tree is constrained, and unlabelled problems where only the number of these vertices is constrained. A variety of labelled and unlabelled problems are originated by considering different tree cost functions.

1. INTRODUCTION

The theory of computational complexity [1,2,3] has provided fundamental insights into the inherent difficulty of many combinatorial optimization problems. Aim of the authors is to classify tree problems from the point of view of their complexity, and to identify the borderline between easy and NP-complete problems.

In a recent work [4] a first set of results concerning spanning tree problems has been presented. In this paper we investigate the computational complexity of Steiner-like problems, i.e. problems asking for the existence in an undirected graph of a tree with bounded cost and constrained set of vertices. If the identity of the vertices of such a tree is constrained we talk about labelled problems, whereas if only the number of these vertices is constrained we speak of unlabelled problems.

Let $\mathcal{T}$ be the set of all pairs (T,U), such that T is tree of an arc weighted undirected graph and U is a subset of V(T), the set of vertices of T. Let $\mathbb{Z}$ and $\mathbb{N}$ denote respectively the set of integers and of natural numbers.

All the labelled (resp. unlabelled) problems here considered are identified by specifying a triple $\langle C, R, \Leftrightarrow \rangle$, where:

- $C : \mathcal{T} \to R$ is a tree weight function
- $R \in \{\mathbb{Z}, \mathbb{N}\}$
- $\Leftrightarrow$ is a relation symbol belonging to the set $\{\subseteq, =, \supseteq\}$ (resp. $\{\leq, =, \geq\}$).

The general problem $\langle C, R, \Leftrightarrow \rangle$ can be formulated in recognition form as follows:

Input: - an undirected graph $G = (V,A)$
- an arc weight function $w : A \to R$
- a constant $W \in R$
- a subset $S \subseteq V$ (resp. a positive integer k)
- a vertex $\rho \in S$.

Property: There exists a tree T of G, containing at least one arc, s.t.
- $V(T) \Leftrightarrow S$ (resp. $|V(T)| \Leftrightarrow k$)
- $C(T,U) \leq W$

where $U = V(T) \cap S$ (resp. $U = V(T)$)

The weight functions considered throughout this paper are natural extentions of those studied in [4]. In fact in most practical labelled problems the vertices of $V(T) \cap S$ play a special role in determining the weight of a tree T, whereas in unlabelled problems all vertices of $V(T)$ evenly contribute to this weight. In the following section we recall these functions and give some general notations. In section 3 we prove some general reductions and in section 4 we present the results concerning the complexity of labelled and unlabelled problems.

2. WEIGHT FUNCTIONS

Before giving the formal definitions of the weight functions, we introduce some general notations. Given a graph $G = (V,A)$ and a subset $X \subseteq V$, we denote by G_X the subgraph of G induced by X.

Given two distinct vertices v_1 and v_2 of a tree T, let $\pi(v_1,v_2,T)$ be the path in T connecting v_1 and v_2 and let

$$p(v_1,v_2,T) = \sum_{a\in\pi(v_1,v_2,T)} w(a) .$$

Moreover, let $p(v,v,T) = 0$ for any vertex v of T and let $p(v_1,v_2,T) = +\infty$ whenever T does not contain both v_1 and v_2.

Given two subsets S_1, S_2 of $V(T)$, the vertex set of a tree T, and an arc a of T, we indicate with $f_{S_1}^{S_2}(a,T)$ the number of paths in T containing arc a and connecting a vertex in S_1 and a vertex in S_2.

Table 1 lists all the tree weight functions considered in this study.

Table 1

Tree weight functions

Expression for C(T,U)	Function name
$\max_{a\in T} w(a)$	MAX ARC
$\sum_{a\in T} w(a)$	SUM ARC
$\max_{a\in T} w(a)\, f_\rho(a,T)$	MAX ROOTED FLOW
$\sum_{a\in T} w(a)\, f_\rho(a,T) \ (= \sum_{v_i\in U} p(\rho,v_i,T))$	SUM ROOTED FLOW (SUM ROOTED PATH)
$\max_{a\in T} w(a)\, f(a,T)$	MAX FLOW
$\sum_{a\in T} w(a) f(a,T) \ (= \sum_{v_i,v_j\in U} p(v_i,v_j,T))$	SUM FLOW (SUM PATH)
$\max_{a\in T} [f(a,T)-w(a)]$	MAX DIFFERENTIAL FLOW
$\max_{a\in T} [f_\rho(a,T)-w(a)]$	MAX ROOTED DIFFERENTIAL FLOW
$\max_{v_i\in U} p(\rho,v_i,T)$	MAX ROOTED PATH
$\max_{v_i,v_j\in U} p(v_i,v_j,T)$	MAX PATH

In this table and throughout the paper $f(a,T)$ and $f_{\rho}(a,T)$ are short hand notations for $f^{U}_{U}(a,T)$ and $f^{U}_{\{\rho\}}(a,T)$ and are called respectively the flow and the rooted flow through arc a in T.

As already mentioned in [4] weight functions involving flows or rooted flows are typical of problems concerning the minimization of communication costs between vertices of U; the last two weight functions are typical of problems where the maximum weighted distance between vertices of U is of concern as for instance in emergency or highly reliable networks. In the fourth and sixth row of the table we have reported in parenthesis an equivalent expression and name for the corresponding tree weight function.

3. GENERAL REDUCTIONS

Throughout this paper, we shall adopt the following formal convenction in order to exhibit polynomial reductions between Steiner-like problems. When showing that P_1 polynomially reduces to P_2 (s.h.n. $P_1 \propto P_2$), the corresponding instances are always denoted as in the general formulation given in section 1; in order to distinguish between the items of these instances, items identifying the instance of P_2 will always be primed.

A set of general polynomial reductions can be established irrespective of the tree weight function and the range R considered. The first two reductions concern labelled problems, reductions 3 and 4 concern unlabelled problems and the last four relate labelled to unlabelled problems.

Reduction 1. $\langle C,R,\equiv\rangle \propto \langle C,R,\supseteq\rangle$.

Proof. $G' = G_S$, $w'(a) = w(a)\ \forall\ a \in A'$, $W' = W$, $S' = S$; $\rho' = \rho$, $T' = T$.

Reduction 2. $\langle C,R,\subseteq\rangle \propto\ C,R,\supseteq\rangle$.

Proof. $G' = G_S$, $w'(a) = w(a)\ \forall\ a \in A'$, $W' = W$, $S' = \emptyset$, $\rho' = \rho$, $T' = T$.

Reduction 3. $\langle C,R,\geq\rangle \propto \langle C,R,=\rangle$.

Proof. The proof is based on the fact that any polynomial algorithm for solving $\langle C,R,=\rangle$ can be used successively with $k' = k, k+1, \ldots, |V|$ to solve $\langle C,R,\geq\rangle$.

Reduction 4. $\langle C,R,\leq\rangle \propto \langle C,R,=\rangle$.

Proof. Similar to that of Reduction 3.

Reduction 5. $\langle C,R,\equiv\rangle \propto \langle C,R,\geq\rangle$.

Proof. $G' = G_S$, $w'(a) = w(a)\ \forall\ a \in A'$, $W' = W$, $k' = |S|$, $\rho' = \rho$, $T' = T$.

Reduction 6. $\langle C,R,\equiv\rangle \propto \langle C,R,=\rangle$

Proof. Same as for reduction 5.

Reduction 7. $\langle C,R,\subseteq\rangle \propto \langle C,R,\leq\rangle$

Proof. Same as for reduction 5.

Reduction 8. $\langle C,R,\subseteq\rangle \propto \langle C,R,\geq\rangle$

Proof. $G' = G_S$, $w'(a) = w(a)\ \forall\ a \in A'$, $W' = W$, $k' = 0$, $\rho' = \rho$, $T' = T$.

All the above reductions are schematically represented by arrows in fig. 1.

4. COMPLEXITY RESULTS

The complexity results obtained are reported in table 2 for what concerns labelled problems and in table 3 for unlabelled problems. In these tables the symbols *,!,? denote respectively a polynomial bounded, NP-complete and undecided problem.

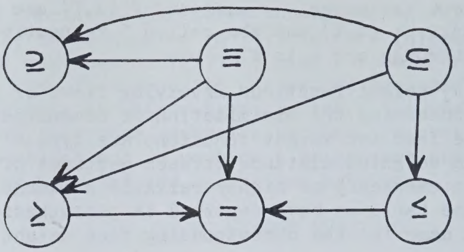


Fig. 1

Table 2
Complexity of labelled problems

Relation symbols	⊇		≡		⊆
Range / Tree weight function	$\mathbb{N}$	$\mathbb{Z}$	$\mathbb{N}$	$\mathbb{Z}$	$\mathbb{Z}$
MAX ARC	*	*	*	*	*
SUM ARC	!	!	*	*	!
MAX ROOTED FLOW	!	!	!	!	!
SUM ROOTED FLOW	*	!	*	!	!
MAX FLOW	!	!	!	!	?
SUM FLOW	!	!	!	!	!
MAX DIFFERENTIAL FLOW	!	!	!	!	*
MAX ROOTED DIFF. FLOW	!	!	!	!	*
MAX ROOTED PATH	*	!	*	!	*
MAX PATH	*	!	*	!	*

It is worth noting that in the last column of both tables the range $\mathbb{N}$ has not been considered. In fact, irrespective of the tree weightings here considered, the corresponding problems reduce to the trivial case where T is constrained to contain only one arc.

In order to prove the results given in the tables, note that the Steiner-like problems having ≡ as relation symbol are equivalent to unconstrained spanning tree problems, whose complexity has already been studied in [4]. Moreover, all problems having MAX ARC as weight function are obviously easy, since they can be solved by finding the connected components of the given graph.

The rest of this section is devoted to the proof of the remaining results contained in the tables.

Table 3

Complexity of unlabelled problems

Relation symbols	$\geq$		$=$		$\leq$
Tree weight function \ Range	$\mathbb{N}$	$\mathbb{Z}$	$\mathbb{N}$	$\mathbb{Z}$	$\mathbb{Z}$
MAX ARC	*	*	*	*	*
SUM ARC	!	!	!	!	!
MAX ROOTED FLOW	!	!	!	!	!
SUM ROOTED FLOW	*	!	*	!	!
MAX FLOW	!	!	!	!	?
SUM FLOW	!	!	!	!	!
MAX DIFFERENTIAL FLOW	!	!	!	!	*
MAX ROOTED DIFF. FLOW	!	!	!	!	*
MAX ROOTED PATH	*	!	*	!	*
MAX PATH	*	!	*	!	*

4.1 Labelled problems

It is well known [1] that $\langle \text{SUM ARC}, \mathbb{N}, \supseteq \rangle$ and therefore $\langle \text{SUM ARC}, \mathbb{Z}, \supseteq \rangle$ are NP-complete. For what concerns the remaining problems with relation symbol $\supseteq$, we need only to prove the results on $\langle \text{MAX ROOTED PATH}, \mathbb{N}, \supseteq \rangle$, $\langle \text{SUM ROOTED PATH}, \mathbb{N}, \supseteq \rangle$, $\langle \text{MAX PATH}, \mathbb{N}, \supseteq \rangle$, since the other results are immediate consequence of [4] and of reduction 1.

(A) $\langle \text{MAX ROOTED PATH}, \mathbb{N}, \supseteq \rangle$ is easy.

(B) $\langle \text{SUM ROOTED PATH}, \mathbb{N}, \supseteq \rangle$ is easy.

Proof. In order to solve both problems it is enough to seek for a tree of least weighted paths from ρ to all vertices of S.

(C) $\langle \text{MAX PATH}, \mathbb{N}, \supseteq \rangle$ is easy.

Proof. Given an arc weighted tree T, an arc $a = \{v_i, v_j\}$ of T, $i < j$, and given an integer x, $0 \leq x \leq w(a)$, $w(a)$ being the weight of arc a, we denote by $T_{a,x}$ the arc weighted tree obtained from T by

(i) adding a new vertex v_a and

(ii) substituting arc a with two arcs $a' = \{v_i, v_a\}$, $a'' = \{v_a, v_j\}$ with weights $w(a') = x$, $w(a'') = w(a) - x$.

Consider the following problem

STEINER CENTRE LOCATION

Input - an undirected graph $G' = (V', A')$
- an arc weight function $w' : A' \rightarrow \mathbb{N}$
- a constant, W'

- a subset $S' \subseteq V'$

Property There exists

- a tree T' of G' with $V(T') \supseteq S'$
- an arc a' of T'
- a non negative integer $x' \leq w'(a')$ s.t.

$$\max_{v_i \in S'} \; p(v_{a'}, v_i, T'_{a'}, x') \leq W' .$$

This problem is the recognition version of a similar optimization problem, referred to as ABSOLUTE CENTRE LOCATION in [5], and can be solved in polynomial time via a straightforward variant of Hakimi's algorithm. In this variant, for each arc a of the graph, and for each location of a new vertex v_a on a, we seek for the shortest paths connecting v_a only to the vertices of S. Moreover, $\langle$MAX PATH, $\mathbb{N}$, $\supseteq\rangle$ can be reduced to STEINER CENTRE LOCATION in a way similar to that used in [4] to reduce $\langle$MAX PATH, $\mathbb{N}$, $\equiv\rangle$ to ABSOLUTE CENTER LOCATION.

Finally, we prove the complexity results of the problems with relation symbol $\subseteq$.

(D) $\langle$MAX ROOTED DIFF. FLOW, $\mathbb{Z}$, $\subseteq\rangle$ is easy.

(E) $\langle$MAX DIFF. FLOW, $\mathbb{Z}$, $\subseteq\rangle$ is easy.

Proof. The answer to problem (D) (resp. (E)) is yes iff there exists an arc $\{\rho,v\}$, $v \in S$ (resp. $\{v_1,v_2\}, v_1,v_2 \in S$) whose weight is not less than 1-W.

(F) $\langle$MAX ROOTED PATH, $\mathbb{Z}$, $\subseteq\rangle$ is easy.

(G) $\langle$MAX PATH, $\mathbb{Z}$, $\subseteq\rangle$ is easy.

Proof. The answer to problem (F) (resp. (G)) is yes iff there exists an arc $\{\rho,v\}$, $v \in S$ (resp. $\{v_1,v_2\}$, $v_1,v_2 \in S$) whose weight is not greater than W.

(H) $\langle$SUM ARC, $\mathbb{Z}$, $\subseteq\rangle$ is NP-complete.

Proof. $\langle$SUM ARC, $\mathbb{Z}$, $\supseteq\rangle$ reduces to it. In fact, assume wlog that $S = \{v_1,v_2,\dots,v_{|S|}\}$ and let

$$V' = V \cup \{u_1,u_2,\dots,u_{|S|}\},$$

$$\bar{A} = \{\{v_1,u_1\},\{v_2,u_2\}, \dots, \{v_{|S|},u_{|S|}\}\},$$

$$A' = A \cup \bar{A}$$

$$G' = (V',A') ,$$

$$w'(a') = \begin{cases} w(a') & \text{if} \quad a' \in A \\ -M & \text{otherwise,} \end{cases}$$

where M is a suitably large positive integer. Let moreover $W' = W - M \cdot |S|$, $S' = V'$, $T' = T \cup \bar{A}$.

(I) $\langle$MAX ROOTED FLOW, $\mathbb{Z}$, $\subseteq\rangle$ is NP-complete.

Proof: $\langle$MAX ROOTED FLOW, $\mathbb{Z}$, $\equiv\rangle$ reduces to it. In fact let $V' = V \cup \{\rho'\}$, $A' = A \cup \{\{\rho, \rho'\}\}$, $G' = (V',A')$, and

$$w'(a) = \begin{cases} |S| \cdot w(a) & \text{if} \quad a \in A \\ W & \text{if} \quad a = \{\rho,\rho'\}. \end{cases}$$

Moreover let $W' = W|S|$, $S' = S \cup \{\rho'\}$, $T' = T \cup \{\rho, \rho'\}$. Assume wlog that $W < 0$ and remember that we look for trees containing at least one arc. Assume T spans S and satisfies $w(a)\ f_\rho\ (a,T) \leq W$ for all $a \in T$.

By construction $f_\rho(a,T) = f_{\rho'}(a,T')$ if $a \neq \{\rho,\rho'\}$ and $f_{\rho'}(\{\rho,\rho'\}, T')=|S|$. It is then sufficient to multiply both sides of the previous inequality by $|S|$, to verify that T' also spans S and satisfies $w'(a)f_{\rho'}(a,T') \leq W'$ for all $a \in T'$. Viceversa let T' be a tree of G' having set of vertices $V(T') \subseteq S'$, root ρ' and including at least one arc; since

$$w'(\{\rho,\rho'\}) \cdot f_{\rho'}(\{\rho,\rho'\}, T') \leq W'$$

we can derive that

$$f_{\rho'}(\{\rho,\rho'\}, T') \geq |S|$$

and therefore that T spans S and $w(a)\, f_\rho(a,T) \leq W$ for all $a \in T$.

(J) $\langle$SUM ROOTED FLOW, $\mathbb{Z}$, $\subseteq\rangle$ is NP-complete.

(K) $\langle$SUM FLOW, $\mathbb{Z}$, $\subseteq\rangle$ is NP-complete.

Proof. $\langle$SUM ROOTED FLOW, $\mathbb{Z}$, $\equiv\rangle$ reduces to $\langle$SUM ROOTED FLOW, $\mathbb{Z}$, $\subseteq\rangle$ (resp. $\langle$SUM FLOW, $\mathbb{Z}$, $\equiv\rangle$ reduces to $\langle$SUM FLOW, $\mathbb{Z}$, $\subseteq\rangle$). In fact, let $\bar{A}$, G', $w' : A' \to \mathbb{Z}$, T' be defined as in the proof of (H). Moreover, let $\rho' = \rho$, $S' = S \cup \{u_1, u_2, \ldots, u_{|S|}\}$ and $W' = 2W - M \cdot |S|$ (resp. $W' = 4W - M \cdot |S|(2|S|-1)$). These reductions are based on the following equalities. For each arc a of T' such that $a \in A$ we have:

$$f_{\rho'}(a,T') = 2\, f_\rho(a,T)$$

(resp. $f(a,T') = 4f(a,T)$).

For each arc $a \in \bar{A}$ we have

$$f_{\rho'}(a,T') = 1$$

(resp. $f(a,T') = 2\,|S|-1$).

The complexity of $\langle$MAX FLOW, $\mathbb{Z}$, $\subseteq\rangle$ is unknown to the authors at present time.

4.2 Unlabellel problems

We consider first SUM ARC tree weightings.

(L) $\langle$SUM ARC, $\mathbb{N}$, $=\rangle$ is NP-complete

Proof. Since $\langle$SUM ARC, $\mathbb{N}$, $\supseteq\rangle$ is NP-complete, we need only to show that $\langle$SUM ARC, $\mathbb{N}$, $\supseteq\rangle$ reduces to $\langle$SUM ARC, $\mathbb{N}$, $=\rangle$.

Assume wlog that $S = \{v_1, v_2, \ldots, v_{|S|}\}$. Let V', $\bar{A}$, A' and G' be defined as in the proof of (H), whereas

$$w'(a') = \begin{cases} w(a') + M & \text{if } a' \in A \\ 0 & \text{otherwise} \end{cases}$$

where M is a suitably large positive integer. Suppose now that there exists a polynomial algorithm for solving $\langle$SUM ARC, $\mathbb{N}$, $=\rangle$. Using this algorithm successively with inputs G', $w' : A' \to \mathbb{N}$, $W' = W+M(|S|-1+j)$, $k' = 2\,|S| + j$, $j = 0,1,\ldots, |V| - |S|$, we would be able to solve $\langle$SUM ARC, $\mathbb{N}$, $\supseteq\rangle$ in polynomial time. In fact being M suitably large, there exists a tree T of G s.t. $V(T) \supseteq S$, $\sum_{a \in T} w(a) \leq W$ and $|V(T)| = |S|+j$ iff there exists a tree $\bar{T} = T \cup \bar{A}$ of G' s. t. $\sum_{a \in T'} w'(a) \leq W+M(|S|-1+j)$ and $|V(T')| = 2|S|+j$.

An immediate consequence of (L) is that $\langle$SUM ARC, $\mathbb{Z}$, $=\rangle$ is NP-complete. Another consequence of (L) is the following

(M) $\langle$SUM ARC, $\mathbb{N}$, $\geq\rangle$ is NP-complete.

Proof. $\langle$SUM ARC, $\mathbb{N}$, $=\rangle$ reduces to it. It is sufficient to observe that

if there exists a tree T s.t. $|V(T)| \geq k$ and $C(T,U) \leq W$, for any subtree T' of T s.t. $|V(T')| = k$, we have $C(T',U) \leq C(T,U) \leq W$.
All the remaining NP-completeness results of table 3 are consequences of those of table 2 and reductions 5,6,7,8.

We now turn our attention to easy problems.

(N) $\langle \text{SUM ROOTED PATH}, \mathbb{N}, = \rangle$ is easy.

(O) $\langle \text{MAX ROOTED PATH}, \mathbb{N}, = \rangle$ is easy.

Proof. It is well known [6] that (i) one can find in polynomial time the list $\pi_1, \pi_2, \ldots, \pi_{|V|-1}$ of the shortest paths from ρ to the other vertices of G in order of non decreasing weight and (ii) any set of these paths forms a tree of G.

Any instance of $\langle \text{SUM ROOTED PATH}, \mathbb{N}, = \rangle$ (resp. $\langle \text{MAX ROOTED PATH}, \mathbb{N}, = \rangle$) is a yes instance iff $\sum_{j=1}^{k} p_j \leq W$ (resp. $p_k \leq W$), p_j being the weight of path π_j.

(P) $\langle \text{MAX PATH}, \mathbb{N}, = \rangle$ is easy.

Proof. Consider the following problem.

CONSTRAINED CENTRE LOCATION (CCL)

Input
- an undirected graph $G' = (V',A')$
- an arc weight function $w' : A' \to \mathbb{N}$
- two constant $k', W' \in \mathbb{N}$

Property there exist
- a tree T' of G' with $|V(T')| = k'$
- an arc a' of T'
- a non-negative integer $x' \leq w'(a')$ s.t. $\max\limits_{v_i \in V(T')} p(v_{a'}, v_i, T'_{a',x'}) \leq W'$,

where $v_{a'}$ and $T_{a',x'}$ are defined as in the proof of (C).
Here again we can refer the reader to [4] in order to obtain a proof for the reduction of $\langle \text{MAX PATH}, \mathbb{N}, = \rangle$ to CCL. Hence (P) is proved if we can find a polynomial algorithm for CCL.

Such an algorithm can be obtained from Hakimi's [5] algorithm for the ABSOLUTE CENTRE LOCATION (ACL) by the following observations. Recall that a key step in Hakimi's algorithm consists in constructing, for each arc $a' = \{v_i, v_j\}$ and for each vertex v_h of the given graph G' a diagram D_h plotting the shortest distance from $v_{a'}$ to v_h, versus x'. For each candidate arc a', the best value of x' both for ACL and CCL, corresponds to the intersection of two diagrams D_h, D_ℓ. In order to solve CCL, for each x' corresponding to an intersection, we compute the shortest paths from $v_{a'}$ to the k' nearest vertices of G'. Since there are at most $O(n^2)$ intersections for each a', it is possible to enumerate in polynomial time all candidate solutions, thus solving CCL.

As a consequence of (N), (O), (P) and reductions 3, 4 the remaining results concerning the tree weightings SUM ROOTED FLOW, MAX ROOTED PATH and MAX PATH follow.

The complexity of $\langle \text{MAX FLOW}, \mathbb{Z}, \leq \rangle$ is still unknown to the authors.
Finally, problems $\langle \text{MAX DIFF. FLOW}, \mathbb{Z}, \leq \rangle$ and $\langle \text{MAX ROOTED DIFF. FLOW}, \mathbb{Z}, \leq \rangle$ are trivially seen to be easy by the same argument used in (D) and (E).

ACKNOWLEDGMENT

This work has been partially supported by the Ministry of Education and by the National Research Council of Italy.

REFERENCES

[1] Karp R.M., "Reducibility among combinatorial problems" in: R.E. Miller and J.W. Thatcher (eds.), Complexity of Computer Computations, Pergamon Press, Oxford & N.Y. (1972) 85-103.

[2] Aho A.V., Hopcroft J.E. and Ullman J.D., The Design and Analysis of Computer Algorithms, Addison Wesley, Reading, Mass. (1974).

[3] Garey M.R. and Johnson D.S., Computers and Intractability, W.H. Freeman & Co., S. Francisco (1979).

[4] Camerini P.M., Galbiati G., and Maffioli F., "Complexity of Spanning Tree Problems: part I", EJOR (to appear).

[5] Christofides N., Graph theory: an Algorithmic Approach, Academic Press, London (1975) Ch. 5.

[6] Dijkstra E., "A Note on two Problems in Connection with Graphs" Numerical Mathematics 1 (1959) 269-271.

REAL AND COMPLEX TENSORS OF NEAR MAXIMAL RANK

CHARLES M. FIDUCCIA
GE Research and Development Center
Schenectady, New York 12345

ABSTRACT

The computational complexity of multiplying two generic elements of a finite-dimensional algebra, such as matrices, is determined by the rank of the tensor of structure constants of the algebra. We contribute to the understanding of the rank function by showing how to construct specific instances of real and complex tensors having nearly any rank desired. The construction is based on Lindemann's theorem and the Eisenstein irreducibility criterion.

INTRODUCTION

The outstanding problem in the area of algebraic complexity is the matrix multiplication problem. Since its asympotic behavior is determined by the number of multiplications used by the algorithm, attention has focused on the minimal number of multiplications required. This number has been characterized [2,3,7] as a purely algebraic quantity, called the rank, of the tensor associated with the multiplication problem. Thus the complexity of matrix multiplication could be determined exactly if an algorithm for computing tensor rank were available.

The quest for a tensor rank algorithm is hampered by our lack of knowledge concerning the behavior of the rank function. Its value has been determined for only a relatively few tensors. It was not until recently that the existence [1] of $n \times n \times n$ tensors of rank $\geq n^2/3$ was established, and that such instances were constructed [5]. This paper further contributes to our understanding of the rank function by showing how to construct specific instances of tensors of nearly any rank over the fields of real and complex numbers. For example, tensors of near maximal rank can be constructed all of whose entries have absolute value =1. It is conjectured that our construction yields tensors of maximal rank, thus providing a better understanding of the rank function.

TENSOR RANK

By a p^{th}-order tensor T of size $n_1 \times n_2 \times \ldots \times n_p$ over a field F we mean a p-dimensional array $T = (\gamma_{i_1,i_2,\ldots,i_p})$ whose entries are in F. We call T simple if there exist vectors $v_j \in F^{n_j}$ $(j=1,\ldots,p)$ such that

$$\gamma_{i_1,i_2,\ldots,i_p} = v_{1,i_1} \cdot v_{2,i_2} \cdots v_{p,i_p} \quad (i_j=1,\ldots,n_j) \qquad (1)$$

which can also be expressed as

$$T = v_1 \otimes v_2 \otimes \cdots \otimes v_p .$$

The rank (over F) of any T is defined [2,3,7] to be the smallest r for which T can be written as a sum $T=T_1+\ldots+T_r$, where each T_k is simple, of the same size as T, and addition is by components.

Third-order tensors are of particular importance in the area of algebraic complexity of computations since it is known [2,3,7] that the number of nonscalar multiplications which are necessary and sufficient to evaluate a bilinear mapping $\mu : F^m \times F^n \to F^r$, in the class of bilinear algorithms, is precisely equal to the rank of its associated $m \times n \times r$ tensor $T=(\gamma_{ijk})$ of structure constants, defined by

$$z = \mu(x,y) \quad z_k = \sum_{ij}\sum \gamma_{ijk} x_i y_j .$$

Of particular interest is the n×n×n case; for then the pair $A = (F^n, \mu)$ is a (generally non-associative) finite-dimensional linear algebra over F, and the rank of T can be related to various abstract properties of A [4].

No general theory currently exists for computing, or even tightly bounding, the rank of a given tensor T. For n×n×n tensors, the bounds $0 \le \text{rank}(T) \le n^2$ follow from the definition. The upper bound can be further improved [6] to

$\frac{3}{4}n^2$. The best known lower bounds on particular tensors are only linear in n, in spite of the fact that most n×n×n tensors are known to have rank $\geq \frac{1}{3}n^2$ [1,6].

TENSORS OF HIGH RANK

The existence of tensors of high rank can be established [1,6] in several ways depending on the domain from which the entries of T are chosen. A generic tensor of high rank (see Corollary 1) was exhibited in [5]. The purpose of this paper is to display specific instances of tensors of high rank over the fields of real and complex numbers.

Let F be any field whose degree of transcendence over its prime field P is sufficiently large, and let $T = (\gamma_{i_1,i_2,\ldots,i_p})$ be an $n_1 \times n_2 \times \ldots \times n_p$ tensor over F. Define the transcendence transc(T) of T as the number of entries of T which are algebraically independent over P, i.e. the transcendence of the field P(T) generated by the entries of T.

Theorem 1. $\operatorname{rank}(T) \geq \dfrac{\operatorname{transc}(T)}{(n_1+n_2+\ldots+n_p)}$.

Proof. If rank(T) = r, T is the sum $T_1+\ldots+T_r$ of r simple tensors. From equation 1, each entry of each $T_k = v_{k1} \otimes \ldots \otimes v_{kp}$ is a product of p of the $N = n_1+\ldots+n_p$ entries of the p factors of each T_k. Thus each entry of T is a polynomial (with zero-one coefficients) of the rN entries of all the rp factors of $T_1,\ldots,T_r$ and is a fortiori an element of the subfield $P(v_{11},\ldots,v_{rp})$ of F generated by these rN entries. This shows that the degree of transcendence, d, of this subfield is in the range: $\operatorname{transc}(T) \leq d \leq rN$. Thus $r \geq \operatorname{transc}(T)/N$.

Corollary 1 [5]. Let G be any field and let $x_{i_1,i_2,\ldots,i_p}$ $(i_j=1,\ldots,n_j)$ be Πn_j independent indeterminates; then over any field F which contains these indeterminates and the prime subfield of G, the tensor $X = (x_{i_1, i_2, \ldots, i_p})$ has $\text{rank}(X) \geq \Pi n_j / \Sigma n_j$.

By counting more carefully, in the proof of Theorem 1, it is possible to establish a slightly better upper bound and hence a better lower bound on rank(T). As it stands, it shows how to construct $n \times n \times n$ tensors of rank $\geq n^2/3$. Regardless of what the exact rank of X is, we conjecture that X has maximal rank among all $n_1 \times \ldots \times n_p$ tensors over F.

The need to concoct the field F in order to display an instance X of high rank leaves something to be desired. A more interesting T would be one over the complex field ₵. Since ₵ has infinite degree of transcendence over its prime field Q of rationals, the theorem tells us how to construct complex tensors of high rank:

Corollary 2. Let $\tau_1, \ldots, \tau_m$ be any m algebraically independent transcendental numbers; then any $n_1 \times n_2 \times \ldots \times n_p$ tensor T which contains $\tau_1, \ldots, \tau_m$ among its entries (without regard to order or frequency) has $\text{rank}(T) \geq m/(n_1 + n_2 + \ldots + n_p)$.

This reduces the problem of exhibiting a tensor of rank $\text{rank}(T) \geq R$ to exhibiting $m = R\Sigma n_j$ independent transcendental numbers. By choosing $m = \Pi n_j$ we may thus exhibit specific instances of complex tensors of rank $\geq \Pi nj/\Sigma nj$. Now, exhibiting $\tau_1, \ldots, \tau_m$ appears to be easier said than done; however, methods of generating arbitrarily long lists of independent transcendental numbers do exist.

LISTS OF INDEPENDENT TRANSCENDENTALS

Transcendental numbers can be generated as values of the exponential function $w = e^z$ by appropriate choices of z. Indeed, we can generate lists of independent transcendentals using:

Lindemann's Theorem [9,11] If $\alpha_1, \ldots, \alpha_m$ are algebraic numbers which are linearly independent over Q, then $e^{\alpha_1},\ldots,e^{\alpha_m}$ are algebraically independent over the field of algebraic numbers.

The task of constructing complex tensors of high rank is thus further reduced to exhibiting a list $\alpha_1,\ldots,\alpha_m$ of linearly independent algebraic numbers. Any m elements of a basis of an algebraic field extension $Q(\alpha)$ will serve this purpose. In particular, it suffices to exhibit an algebraic number α of degree $d \geq m$ and take $\alpha_k = \alpha^k$ and $\tau_k = e^{\alpha^k}$ $(k=0,1,\ldots,m-1)$. Any root α of an irreducible polynomial $\varphi(x) \in Q[x]$ of degree d will serve this purpose since $1,\alpha,\alpha^2,\ldots,\alpha^{d-1}$ constitute a basis for $Q(\alpha) \simeq Q[x]/(\varphi(x))$.

We have thus reduced our original tensor problem to finding an irreducible polynomial $\varphi(x)$ of degree $d \geq m$ over Q. Such polynomials may be found with the help of the Eisenstein irreducibility criterion [8]. Combining Lindemann's Theorem with Corollary 2, we obtain:

Theorem 2. Let $\alpha \in \mathbb{C}$ be any root of any irreducible polynomial $\varphi(x) \in Q[x]$ of degree $d \geq m$, and let $\alpha_1,\ldots,\alpha_m$ be any m elements of a basis for $Q(\alpha)$ over Q, such as $1,\alpha,\alpha^2,\ldots,\alpha^{d-1}$. Then any complex tensor T of size $n_1 \times n_2 \times \ldots \times n_p$ which contains $e^{\alpha_1},\ldots,e^{\alpha_m}$ among its entries has $\mathrm{rank}(T) \geq m/(n_1+n_2+\ldots+n_p)$, over any field F which contains the subfield $Q(e^{\alpha_1},\ldots,e^{\alpha_m})$.

EXAMPLES

From Theorems 1 and 2 we see that in order to display a specific $n_1 \times n_2 \times \ldots \times n_p$ complex tensor T of rank $(T) \geq R$ we need only exhibit a list of $m = R(n_1+n_2+\ldots+n_p)$ independent transcendental numbers.

$$\tau_k = e^{\alpha^k} \qquad k=0,\ldots,m-1. \tag{2}$$

As long as $\tau_0,\ldots,\tau_{m-1}$ are included in T, without regard to order or frequency, the tensor T will have $\mathrm{rank}(T) \geq R$. If desired, we can construct T so that its rank is order R, as follows: Without loss of generality, assume $n_p = \max(n_1,\ldots,n_p)$. Fill in T vector-by-vector along its p^{th} dimension by using n_p distinct τ_k per vector. The resulting T has

$\mathrm{rank}(T) \le \left\lceil \frac{m}{n_p} \right\rceil \le \lceil pR \rceil$. Since p is a constant independent of R and since $R \le \mathrm{rank}(T) \le \lceil pR \rceil$, we have $\mathrm{rank}(T) = \theta(R)$.

A rich source of irreducible polynomials is the class of cyclotomic polynomials [8-11]. The n^{th} cyclotomic polynomial $\Phi_n(x)$ has as its roots the $\varphi(n)$ primitive n^{th} roots of unity, where $\varphi(n) = n(1-1/p_1)(1-1/p_2)\ldots(1-1/p_r)$ is the Euler φ-function and $p_1,\ldots,p_r$ are the distinct prime divisors of n. Thus, if w is a primitive n^{th} root of unity, then w has order $\varphi(n)$, i.e. $[Q(w):Q] = \varphi(n)$, so that $1,w,w^2,\ldots,w^{\varphi(n)-1}$ are linearly independent over Q. Thus to construct m independent transcendentals we may use any primitive n^{th} root of unity for α, provided $\varphi(n) \ge m$.

Example 1. For any prime $p \ge m+1$, the p^{th} root $w = e^{2\pi i/p}$ has degree $\varphi(p) = p-1 \ge m$. Thus in equation (2) we may take

$$\tau_k = e^{e^{2\pi ik/p}} \qquad k=0,1,\ldots,m-1.$$

Example 2. For any $n = 2^{s+1} \ge 2m$, the n^{th} root $w = e^{\pi i/2^s}$ has degree $\varphi(n) = 2^s \ge m$. Thus in equation (2) we may take

$$\tau_k = e^{e^{\pi ik/2^s}} \qquad k = 0,1,\ldots,m-1.$$

The above examples give complex-valued τ_k with absolute values in the interval $1/e \le |\tau_k| \le e$. The next example gives real-valued τ_k in the same interval.

Example 3. If $w = e^{i\theta}$ is a primitive n^{th} root of unity (of degree $\varphi(n)$), then $w+1/w = 2\cos\theta$ is a real number of degree $\varphi(n)/2$ [see 10], as is the real number $\cos\theta$. Thus for any

$n = 2^{s+1} \geq 4m$, the real number $\cos(\pi/2^s)$ has degree $\varphi(n)/2 = n/4 \geq m$. Thus in equation (2) we may take

$$\tau_k = e^{\cos^k(\pi/2^s)} \qquad k = 0,1,\ldots,m-1.$$

Other real tensors with larger entries may be constructed from a square-free integer q since $x^n - q$ is irreducible for any square-free integer $q \neq \pm 1$ and any n.

Example 4. For any $n \geq m$ and any square-free integer $q \neq \pm 1$ (such as any prime p) the real n^{th} root $\sqrt[n]{q}$ has degree $n \geq m$. Thus in equation (2) we may take

$$\tau_k = e^{q^{k/n}} \qquad k = 0,1,\ldots,m-1.$$

Having constructed both real and complex tensors of high rank, it is possible to transform them into tensors whose elements have some special property, such as all entries lying on a circle or on a line. This is easily done by performing various operations on the algebraic numbers $\alpha_1,\ldots,\alpha_m$ or on the transcendentals $e^{\alpha_1},\ldots,e^{\alpha_m}$. Note that $e^{i\theta}$ is on the unit circle for any real θ. Thus if we desire m independent transcendentals $\tau_0,\ldots,\tau_{m-1}$ on the unit circle, we simply choose a real α of degree $\geq m$ and take $\tau_k = e^{i\alpha^k}$ $(k=0,\ldots,m-1)$. This works because for any algebraic $\beta \neq 0$, the algebraic numbers $\beta\alpha_1,\ldots,\beta\alpha_m$ are linearly independent over Q if and only if $\alpha_1,\ldots,\alpha_m$ are lindearly independent over Q.

Example 5. For any prime p and any $n \geq m$, the numbers $\alpha_k = ip^{k/n}$ $(k=0,1,\ldots,n-1)$ are linearly independent (over Q) imaginary algebraic numbers, so that e^{α_k} are independent transcendentals on the unit circle. Thus in equation (2) we may take

$$\tau_k = e^{ip^{k/n}} \qquad k = 0,1,\ldots,m-1.$$

SUMMARY

Real and complex tensors of any order having nearly any rank can be exhibited by including independent transcendental numbers among their entries. We conjecture that for a given size $n_1 \times n_2 \times \ldots \times n_p$ a tensor of maximal rank is obtained by choosing all Πn_j entries to be independent transcendentals. The arbitrarily large sets of independent transcendentals were constructed using Lindemann's theorem which reduced the problem to finding algebraic numbers linearly independent over Q. These were obtained as the powers of some root α of an irreducible polynomial $\varphi(x)$ whose irreducibility was established using the Eisenstein criterion.

These lower bounds on rank will hopefully lead to some nontrivial lower bounds in the area of algebraic complexity of computations; however, what is really needed are methods for either computing or tightly bounding the rank of a given tensor. Foremost on this list of tensors is the $n^2 \times n^2 \times n^2$ tensor of structure constants of the algebra of $n \times n$ matrices.

REFERENCES

[1] R.W. Brockett, The multiplicative complexity of a family of bilinear forms can be high (abstract), in Algorithms and Complexity (J.F. Traub Ed.), Academic Press, New York, 1976, p. 437.

[2] R. Brockett and D. Dobkin, On the optimal evaluation of a set of bilinear forms, in Proc. 5th Ann. ACM Symp. Theory of Computing, Austin, Texas (1973), pp. 88-95 and in Linear Algebra Appl. 19(1978), pp. 207-235.

[3] C.M. Fiduccia, On obtaining upper bounds on the complexity of matrix multiplication, in Complexity of Computer Computations, (R. Miller and J.W. Thatcher, Eds.), Plenum Press, New York, 1972, pp. 31-40.

[4] C.M. Fiduccia and Y. Zalcstein, Algebras having linear multiplicative complexities J. ACM, 24(1977), pp. 311-331.

[5] C.M. Fiduccia, Hard-to-compute bilinear forms, in Proc. 1978 Conf. Inf. Sciences and Systems, The Johns Hopkins Univ., pp. 40-45.

[6] T.D. Howell, Tensor rank and the complexity of bilinear forms, Ph.D. thesis, Cornell Univ., 1976.

[7] V. Strassen, Vermeidung von Divisionen, J. Reine Angew Math. 264(1973), pp. 184-202.

[8] G. Birkoff and S. MacLane, A Survey of Modern Algebra, Macmillan, New York, 1965, pp. 77-78.

[9] S. Lang, Algebra, Addison-Wesley, Reading, MA, 1965, pp. 493-499.

[10] I. Niven, Irrational Numbers, Wiley and Sons, 1956, pp. 36-38.

[11] C.L. Siegel, Transcendental Numbers, Princeton Univ. Press, Princeton, 1949, pp. 20-24.

MULTIPROCESSING COMPUTER ORGANIZATION WITH MULTIPLE READ SIGNLE WRITE MEMORY

SHELDON S.L. CHANG
Department of Electrical Engineering, SUNY, Stony Brook, N.Y. 11794

Abstract Multiprocessing computer organizations with multiple read single write (MRSW) memory in addition to conventional memories are studied for the following applications: (i) general purpose computing, (ii) scientific computing. Computational effectiveness is improved by reducing or completely eliminating memory and bus conflicts. The processor-memory organization is determined by a module correlation criterion, which also yields a quantitative measure of the residue probability of conflicts.

INTRODUCTION

Research into multiprocessing computer systems has gained momentum with the advent of the microprocessor because of its cost-effectiveness. A computing system which uses a number of microprocessors working cooperatively on a computational task holds the promise of high overall speed, through put, and reliability at low cost.

A number of different system architectures or interconnections of processors, memory, and I/O devices have been proposed.[1,2] The most common structures use one or a small number of time shared buses to interconnect processors to memory and I/O devices. Such systems however suffer from problems due to bus contentions and memory conflicts which degrade the system overall performance.[3-8] The use of a cross bar switch to link dynamically processors and memory modules is one approach which has been pursued to solve the bus contention problem.[9] However the complexity of the cross bar switch increases as N^2 and its cost predominates the system cost as N becomes large. As various structures are devised to handle bus contention and memory conflict problems using single-read single-write memories the resulting structures tend to perform in an optimal manner for only a very limited class of problems. The MRSW memory opens new opportunities to the solution of these problems.

MRSW MULTIPROCESSING ORGANIZATION

The Multiple Read Single Write Memory Fig. 1 represents one bit of a typical memory chip, where Q_1, Q_2, Q_3 and Q_4 constitute a C-MOS flip-flop; Q_7, Q_8, Q_{13} and Q_{14} provide coincidence switching for the READ and WRITE data lines D_o and $\bar{D}_o$; and the remaining transistors provide coincidence switching for four independent READ only data lines: D_1, D_2, $\bar{D}_3$ and $\bar{D}_4$. The sixteen transistors constitute one cell of the memory chip. By making the width to length ratio of the flip-flop transistors large compared to that of the switching transistors, the stored voltages v, $\bar{v}$ are not significantly disturbed by simultaneous reading of a few data lines. The push pull data lines D_o and $\bar{D}_o$ improve the writing speed.

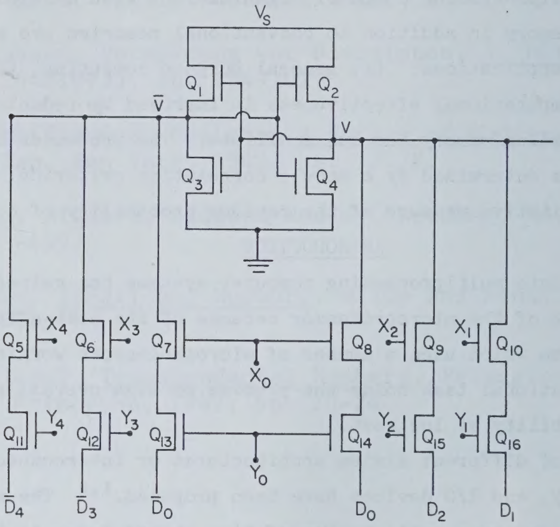


FIG. 1. A CELL OF 1-BIT MEMORY

In the entire chip, there are 5 independent sets of x and y decoders. The data and address lines of each set are connected in the usual manner. The result is a 2^n word memory which can be addressed independently and read simultaneously through the five data lines and written only through the data line $D_o - \bar{D}_o$, where $n = n_x + n_y$ is the total number of address bits.

The MRSW memory is basically advantageous from three considerations:

(1) In most computational programs, the read instruction outnumbers the write instruction.

(2) In a multiprocessing environment, the write-memory space is a subset of its read memory space.

(3) From the point of view of electronics design, switching circuits in the write-line is considerably more expensive than that in a read-only-line for an active memory. The former is a passage for transmitting power to switch a flip-flop circuit. In contrast the latter is required only to transmit signal to the input end of a high gain amplifier, and its "real estate" requirement is considerably less.

Computer Organization with MRSW Memory While MRSW can be used in many ways in a computing system, the present discussion will be restricted to organizations in which a number of MRSW's serve jointly as the "cache" or "buffer" memory in a multiprocessing environment.

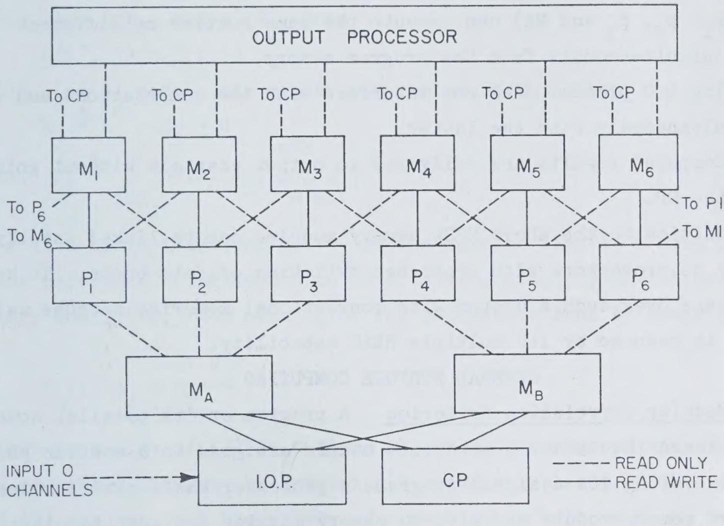


FIG. 2. DISTRIBUTED NETWORK FOR SCIENTIFIC PROCESS.

Fig. 2 shows an organization with a linear array of modular processors $p_1, p_2 \cdots\cdots p_N$ (N = 6 in Fig. 2), interconnected lineally by MRSW memory modules. Each processor can read part of the main memory M_A or M_B and the Central Processor has READ-WRITE capability on the main memory and also reading capability of the module memories. I/O to the main memories is handled either by the CP or separately by an Input Output Processor, while data from the module memories can be connected directly by the Output Processor to output channels and to the main memory.

The following operational characteristics are obvious:

1. As soon as a data point is computed and written into a memory module, it can be immediately read by neighboring processors without the need of data transfer.

2. Each processor has one memory module as its writing space and a number of modules as its accessible reading space. Within these confines the processors operate simultaneously, independently or cooperatively.

3. Communication between the CP and modular processors (MP) can be

accomplished simply by using interrupts. When a MP is interrupted by the CP, control of the MP is transferred to an assigned address in the main memory. When the CP is interrupted by a MP, control of the CP is transferred to an assigned address in the memory module of the MP. Computations by other MP's are not interfered in anyway by the communication process. One significant result is that processors which share program memory (for example $p_1 \cdot p_2$, p_3 and MA) can execute the same routine or different routines simultaneously from the program memory.

4. The I/O process does not interfere with the computations and can be done simultaneously with the latter.

5. Computed results are delivered to output channels without going through the CP.

In addition to the above MRSW memory modules can be linked totally or partially to processors with cross-bar switching or data buses. It holds an advantage over such a system with conventional memories because waiting for MRSW is reduced by its multiple READ capability.

GENERAL PURPOSE COMPUTING

The Modular Correlation Criterion A program or its parallel nodes can be partitioned into modules with each module assigned to a modular processor. In executing its assigned program, a processor makes memory references to its own memory module and also to memory modules assigned to other processors. An instruction (data, I/O) reference is made up of issuing of an address to memory and subsequent fetching of 1 byte of instruction (data, I/O) from memory. The total number of instruction, data, and I/O references will be referred to as the number of memory references in the subsequent analysis.

Let R(i,j) represent the probable number of memory references issued by processor i to memory j in the time T of execution of a program. The total number of references issued by the i-th processor is denoted by $R_p(i)$ and that received by the j-th memory module is denoted as $R_m(j)$:

$$R_p(i) = \sum_j R(i,j) \tag{1}$$

$$R_m(j) = \sum_i R(i,j) \tag{2}$$

The correlation index C(k,j) is defined as

$$C(i,j) \triangleq \frac{R(i,j)}{R_m(j)} \tag{3}$$

The correlation index C(i,j) can be used as a criterion for selecting an efficient processor - memory arrangement. As discussed in the preceding section, each memory module has a host processor which writes on the module. Let h_j denote the host processor, and L_j denote the number of read only ports of memory j. For memory j, the processors are partitioned into three subsets; h_j, R_j, and B_j. Each of the processors in R_j has a read only port in M_j. The remaining N_j read-only ports are shared by the processors in B_j through buses or cross-bar switching. A partition is said to be efficient if the following conditions are satisfied.

(i) Let $i \varepsilon B_j$, and $i' \varepsilon R_j$. Then

$$C(i', j) \geq C(i,j) \tag{4}$$

(ii) Let S_j denote the membership number in R_j. Then:

$$N_j \triangleq L_j - S_j \geq 0 \tag{5}$$

(iii) $$\rho_j \triangleq \sum_{i \varepsilon B_j} C(i,j) \leq \frac{\varepsilon T N_j}{T_m R_m(j)} \tag{6}$$

where ε is a small fraction, and T_m is the service time at a read only port for each reference. The selection of ε is determined by the allowed number of memory conflicts. The meaning of (6) becomes evident by rewriting it as:

$$R_m(j)\, \rho_j \leq \frac{\varepsilon T N_j}{T_m} \tag{7}$$

The right-hand side of (7) is ε times the maximum number of references which can be processed through the remaining N_j read-only lines in the interval T, and (7) assures that the number of remaining references to M_j is less than a small fraction ε of the processable maximum.

The reduction in memory conflict is represented by ρ_j and N_j. With a single read and write memory, every processor belongs to subset B_j and both ρj and N_j equal to 1.

The reduction in memory conflict is said to be complete if $\rho_j = 0$. In scientific computing, it is often possible to obtain a complete solution ($\rho_j = 0$ for all j). It will be discussed in a subsequent section.

Mathematical Analysis of Memory Conflict In general the expected number of memory conflicts is not completely determined by $R_m(j)$, ρ_j, N_j, T, and T_m. The total number of processors and the distribution of memory references among them are also significant factors. A detailed analysis for the single read and write memory case is given by Fung and Torng[7]. The purpose of the present analysis is to obtain an approximate estimate of memory conflicts which can be readily calculated for the general case with $N_j \geq 1$. It is assumed that there are a large number of processors in the subset B_j so that the temporal distribution of memory references approximates Poisson distribution:

$$P(n) = \frac{(\nu T_m)^n}{n!} e^{-\nu T_m} \tag{8}$$

where $\nu = R_m(j)\ \rho_j/T$, and $P(n)$ is the probability of having n references in a memory cycle time T_m. As only N_j references are serviced, the expected number of conflicts is

$$f(\nu T_m, N_j) = \sum_{n=N_j+1}^{\infty} (n-N_j)p(n) = \nu T_m g(N_j) - N_j g(N_j + 1) \tag{9}$$

where

$$g(N_j) = e^{-\nu T_m} \sum_{n=N_j}^{n=\infty} \frac{(\nu T_m)^n}{n!} \tag{10}$$

The function $f(\nu T_m, N_j)$ is plotted in Fig. 3.

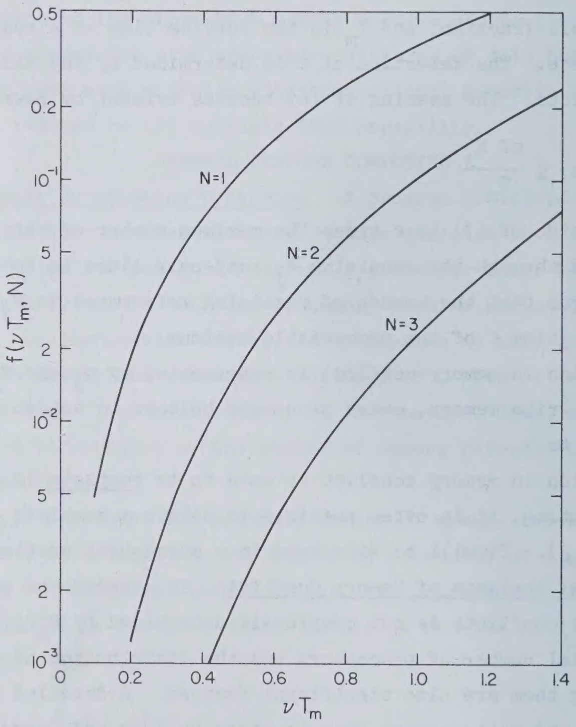


Fig. 3 Memory conflicts versus number of references per cycle

The expected number of memory conflicts from memory j in T is

$$C_j = \frac{T}{T_m} f(\nu T_m, N_j) \tag{11}$$

Let A_i denote a subset of memory modules specified by:

$j \varepsilon A_i$ iff $i \varepsilon B_j$

The expected number of memory conflicts for processor i in T is then

$$E(i) = \sum_{j\varepsilon A_i} \frac{C_j C(i,j)}{\sum_{i'\varepsilon B_j} C(i', j)} \qquad (12)$$

SCIENTIFIC COMPUTING

Partial Differential Equations   Algorithms for computer solution of partial differential equations is a significant area for continued research. The present section will be limited to a special case; the heat transfer problem:

$$\rho C \frac{\partial T}{\partial t} = K\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2}\right) + Q \qquad (13)$$

where Q is the internally generated heat. Let λ and τ denote small lineal and temporal dimensions in which T can be considered uniform. Let x, y, z and t be expressed in units of λ and τ. Eq. (13) can be written as:

$$\begin{aligned} T(\underline{r}, t+1) &= T(\underline{r}, t) + \frac{\tau K}{\lambda^2 \rho C} [\nabla_x^2 T(\underline{r}, t) \\ &\quad + \nabla_y^2 T(\underline{r}, t) + \nabla_z^2 T(\underline{r}, t)] + \frac{\tau}{C} Q(\underline{r}, t) \end{aligned} \qquad (14)$$

where $\underline{r}$ represents x, y, z, and

$$\begin{aligned} \nabla_x^2 T(\underline{r}, t) &\triangleq T(x+1, y, z, t) - 2T(x, y, z, t) \\ &\quad + T(x-1, y, z, t) \end{aligned} \qquad (15)$$

The symbols ∇_y^2, and ∇_z^2 are similarly defined.

Heat flow at the surface can be represented by:

$$\nabla_x^2 T(0,y,z,t) = T(1, y, z, t) - T(0, y, z, t) + \frac{\lambda}{K} S(0, y, z, t) \qquad (16)$$

Equations (14) with ∇^2 defined by (15), (16) and similar equations for y, z are the partial difference equations for the heat transfer problem.

Suppose that a multiprocessing organization is used with each processing element (PE) responsible for computing one line of T: $T(x, y_1, z_1, t)$ where y_1 and z_1 are fixed for each PE, and a memory module is assigned to each PE. With a conventional type of RAM one of the following problems will arise:

(i) If the data on $T(x, y_1, z_1, t)$ is stored only in $M(y_1, z_1)$, the memory module for y_1, z_1, $PE(y_1, z_1)$ must access its own memory and four others: $M(y_1 + 1, z_1)$, $M(y_1 - 1, z_1)$, $M(y_1, z_1 + 1)$ and $M(y_1, z_1 - 1)$ for computation of each point.

(ii) If the data on $T(x, y_1, z_1, t)$ is stored in all five memory modules, there is the problem of updating or memory transfer after each round of computation.

In either case (i) or (ii) there is the potential problem of memory and bus conflicts. In contrast, with MRSW memory modules, one can assign a reading port of $M(Y_1, z_1)$ to each of the four neighboring PE's.

$PE(y_1 \pm 1, z_1)$ and $PE(y_1, z_1 \pm 1)$, and store the data on $T(x, y_1, z_1, t)$ only in $M(y_1, z_1)$. Then each PE has direct access to all the data it needs, and the memory and bus conflict problem is completely solved.

Dynamic Programming is one of the basic methods for optimization. Bellman's principle of optimality can be expressed

$$R(\underline{x}, t) = \underset{\underline{u}}{\mathrm{Max.}}\{g(\underline{x}, \underline{u}, t) + R(\underline{x}(t+1), t + 1)\} \tag{17}$$

$$\underline{x}(t + 1) = \underline{x}(t) + \underline{f}(\underline{x}, \underline{u}, t) \tag{18}$$

where $\underline{x}$ and $\underline{u}$ are state and policy vectors respectively. $R(\underline{x}, t)$ is Bellman's return function; g represents step return; and (18) is the vector state equation. Computation with dynamic programming starts with the last stage return $R(\underline{x}, N)$ and works backwards in t.

In a multiprocessing algorithm, let W_i represents the set of state points assigned to PE_i: PE_i computes $R(\underline{x}_a, t)$ iff $\underline{x}_a \in W_i$. Let R_i represents the set of state points on which PE_i has direct reading access. If

$$\underline{x}_b \in R_i \tag{19}$$

for every $\underline{x}_b$ satisfying

$$\underline{x}_b = \underline{x}_a + \underline{f}(\underline{x}_a, \underline{u}, t)$$

where $\underline{x}_a \in W_i$, then P_i has direct access to all required data for its computations. The condition (19) is easily realized, because $\underline{f}(\underline{x}_a, \underline{u}. t)$ represents the change of state in one step, and it is in general quite samll in magnitude.

For Multi-dimensional Digital Signal Processing a basic algorithm is the frame recursive processing method (FRP)[10]. By repeated application of a small FIR filter (the frame processor) it yields optimum solutions for a large class of problems including image restoration, and tracking of a moving object.

Let W_i and R_i be defined as before. Let $(2d_1+1) \times (2d_2+1) \times \cdots (2d_n+1)$ denote the size of the n-dimensional frame processor. Then P_i has direct access to its required data if $\underline{x}_b \in R_i$ for all $\underline{x}_b$ satisfying

$$|x_{bk} - x_{ak}| \leq d_k \tag{20}$$

and $\underline{x}_a \in W_i$.

CONCLUSIONS

Multiple Read Single Write memory modules are proposed for a basic solution of the memory conflict problem in distributed computing. Module correlation coefficients are introduced as an effective link between hardware and software aspects of the problem. On the software side, module correlation coefficients depend on the computational program, its partition into modules, and the assignments of instructions and initial data to memory modules. On the hardware side, the correlation coefficients determine the multiprocessor-memory organization and its effectiveness in eliminating memory conflicts. Most effective or complete elimination of memory conflict is possible if the correlation coefficients of each memory module have significant value for a few processors only.

The use of MRSW memory opens a new area of investigation for scientific computing algorithms: Parallel processing is at its fastest or most efficient if each P.E. is self-sufficient within its directly accessible memory space.

References

1. G.A. Anderson, and E.D. Jensen, "Computer Interconnection Structures: Taxonamy, Characteristic and Examples", Computing Surveys, Dec. '75 Vol. 7, No. 4.
2. P. H. Enslow, ed., Multiprocessors and Parallel Processing, Wiley, 1974.
3. G. Regling Jr., "Performance and Control of Multiple-Microprocessor Systems", Computer Design, pp. 81-86, March 1974.
4. F. E. Hearst et al., "A new Minicomputer/Multiprocessor for the ARPA Network", in AFIPS Proc. Nat. Computer Conf. and Expos. Vol. 42, pp. 529=537, 1973.
5. V. K. Ravndran and T. Thomas, "Characteristic of Multiple Microprocessor Networks", Proc. IEEE Computer Soc. Int. Conf., pp. 133-137, 1973.
6. D. P. Bhanderkar, "Analysis of Memory Interference in Multiprocessors", IEEE Trans. on Computers, Vol. C-24, pp. 897-908, Sept. 1975.
7. K. T. Fung, and H. C. Torng, "On the Analysis of Memory Conflicts and Bus Contention in a Multiple-Microprocessor System", IEEE Trans. on Computers, Vol. C-28, No. 1 January 1979.
8. G. S. Rao, H. S. Stone, and T. C. Hu, "Assignment of Tasks in a Distributed Processor System with Limited Memory", IEEE Transactions on Computers, Vol. C-28, No. 4, April 1979.
9. W. A. Wulf, and C. A. Bell, "CMMP - A Multiminiprocessor" AFIPS Conf. Proc. Vol. 41, Part II, pp. 765-777, FJCC 1972.
10. S. S . L. Chang, "Digital Linear Processor Theory and Optimum Multidimensional Data Estimation", IEEE Transactions on Automatic Control, Vol. AC-24, No. 2, April 1979.

REVOLUTIONARY DESIGNS FOR ASYNCHRONOUS SEQUENTIAL CIRCUITS USING LSI

GARY K. MAKI and JONICA L. KEEL
Electrical Engineering Department
University of Idaho
Moscow, Idaho 83843

ABSTRACT

A new revolutionary design procedure is presented that will allow the logic designer of the 1980's to be able to implement asynchronous sequential circuits with LSI devices. A large gap presently exists where logic designers are unable to make use of LSI devices because the available design tools were developed a decade ago for random logic. Some authors have attempted to implement circuits with ROMs using the outdated tools and discovered that exceedingly large ROMs are needed to realize even small flow tables. The new design procedure accomplishes the following:

1) Efficient implementation of large flow tables. The hardware requirements in the new approach grow linearly with the size of the flow table whereas the classical approaches grow exponentially by powers of 2. For example, flow tables with up to 1024 stable states per input column can be realized with a few 2708 (1k x 8) ROM bytes. The new procedure is simple and straightforward and does not require the use of state tables or Karnaugh maps.
2) Multiple input variables changes are permitted.
3) Static, dynamic, function, and essential hazards do not cause malfunctions in circuit operation. Classical asynchronous design techniques are replete with hazard problems.
4) Real time fault detection is possible with little added circuitry. The class of fault conditions for which fault detection is guaranteed is that which is associated with a single chip failure; all fault conditions restricted to a single chip, regardless of the number of signals on the chip that are in error or the direction of the error, will be detected.

INTRODUCTION

The tremendous advances of LSI technology has impacted virtually every phase of logic design except that of asynchronous sequential circuits. The design techniques currently available were developed 10 to 15 years ago for random logic using gates and flip-flops. These techniques are of limited use for several reasons: First, these methods are generally implemented by hand thereby being slow and suitable only for small problems. Secondly, the optimization designed into the algorithms has centered around reducing the number of individual gates which is noconcern when using LSI. Finally, state assignment procedures generate an unreasonable number of state variables for even small flow tables.

There are two motivating reasons for developing new design algorithms for asynchronous sequential circuits. The obvious first reason is to reduce the total component count in a circuit and to implement larger flow tables. The second reason is to gain more versatility in a design. Implementing a circuit with a Read-Only Memory (ROM) for example, will allow for reprogramming the ROM to correct a design oversight or to expand the capabilitiy of the circuit to perform additional functions.

Hazards are a phenomena that asynchronous sequential logic designers would like to assume did not exist (the academic design algorithms often do exactly that). The hazards addressed here that plague designers are static, dynamic, functional, and essential hazards [3]. Even if a design algorithm for asynchronous sequential circuits was easy to use and produced minimal hardware (gates or LSI), hazards may still prevent these circuits from being used. Therefore, for a design procedure to be useful in the 1980's, it must successfully resolve the problem of hazards.

Reliability and maintainability are becoming exceedingly important as the logic circuits become more sophisticated. A malfunction in a digital circuit causes losses that range from inconvenience to human life. A digital control system, particularly one that operates in real time, should be designed to be quickly repairable to minimize losses. As the circuits become more sophisticated, the capabilities of the repairman must also become increasingly greater unless the circuit itself has the ability to aid in the diagnosis. Therefore, reliable and easily maintainable designs are important considerations of the future. The circuits designed here have real-time fault detection capability.

Design Procedure

Asynchronous sequential circuits are to be implemented here with a particular LSI device known as a ROM. The inputs and output of a ROM are defined as follows, with Fig. 1 showing the waveforms associated with a tyipcal ROM.

A - Address with n input lines capable of addressing 2^n memory locations

D - Data output lines present the value of a specified memory location

EN - Enable line defined such that when EN=1, internal tri-state buffers are enabled that allow the contents of the memory location specified by A to be placed on D. When EN=0, D assumes the tri-state (high-impedance) value.

Several papers have been presented on the subject of implementing asynchronous sequential circuits using ROMs [1,2]. Figure 2 depicts the general organization which uses feedback loops for address calculation. The procedures of the past may work well for small circuits, however, like the procedures developed for random logic, they fail with larger flow tables. Consider example A to illustrate this point.

Example A: A 20 row flow table with 8 input states (3 input state variables) has 8 stable states in each input. Each stable state presents four output state variables.

If one were to use a Liu [4] state assignment and if the flow table columns were sufficiently unrelated such that state variables could not be shared, then 24 state variables would be needed. Using Sholl's [1] technique, the entire flow table is placed into a ROM, programming it to execute the flow table action. For Example A, a total of 27 inputs would have to be presented to the ROM (24 state variables plus 3 input state variables), and 28 outputs generated (24 state variables plus 4 output state variables). Thus, a $2^{27} \times 28$ ROM would be needed.

The reason for needing the design techniques should be evident; the present methods do not utilize the power of the LSI device, nor are they suitable to implement even moderately small problems. The remainder of the paper presents a new design algorithm taylored for LSI. First, some fundamental definitions are necessary.

A partitioning variable of a flow table column I_p is a state variable that partitions transition paths of different k-sets. A k-set is the set of states in I_p that have the same next state entry. In effect, these are the state variables that are derived from I_p using any asynchronous state assignment technique like that of Liu or Tracey [4,5]. The design equations can take on the following form where I_p is the input state, Y_i the next state variable, Z_j the output state variable, and f_{ip} and g_{jp} are functions of the state variables y_{ip}.

$$Y_i = f_{i1}I_1 + f_{i2}I_2 + \ldots + f_{im}I_m \qquad Z_j = g_{j1}I_1 + g_{j2}I_2 + \ldots g_{jm}I_m \qquad (1)$$

The key to the new design procedure is in the following theorem, which was proven by Sawin and Maki [7].

Theorem A: Any asynchronous sequential circuit coded with a Single Transition Time (STT) state assignment will possess the following property for the coefficients f_{ip} and g_{jp} of Eq. 1: The coefficients f_{ip} and g_{jp} are functions only of the partitioning variables of I_p. Furthermore, the f_{ip} terms associated with the partitioning variables of I_p do not change during a transition.

Consider Example A to illustrate the above theorem. Using a Liu assignment, there are 3 partitioning variables for each column. Let y_1, y_2, y_3 be the partitioning variables for I_1. Theorem A states that all coefficients f_{i1}, $i=1,2,\ldots,24$ and g_{j1}, $j=1,2,3,4$ for the 24 state variables and 4 output states are a function only of y_1, y_2, and y_3.

Theorem A presents the basis for a new design algorithm. Since the next state and output state variables depend only on the partitioning variables of the present input state I_p, it follows that the only memory necessary in the entire circuit is that which is associated with the partitioning variables of I_p. Therefore, the circuit shown in Fig. 3 will provide the necessary circuitry to accomplish this and generate the proper values for all Y_i and Z_j when I_p is present. In Example A, a Liu assignment would require 3 partitioning variables per column; hence a total of 24 state variables. For a given input state, only 8 states of memory are needed to generate the 24 next state variables and 4 output state variables; hence an 8 x 28 ROM is needed for each input in this example.

Consider a small example in Fig. 4 to illustrate further how the overall circuit would function. An acceptable Liu state assignment is shown where each column requires 2 state variables. The partitioning variables for I_1, I_2, and I_3 are y_1y_2, y_3y_4, and y_5y_6 respectively. Figure 5 depicts the general structure of the total circuit implementation for the 3 input states. Each ROM has four sets of outputs, partition variables 1,2,3, (PV1, PV2, PV3) and output Z. Since there are 2 partitioning variables per input, a total of 6 state variables and 2 outputs are needed. Each ROM is 4 x 8 and would be coded (programmed) as shown in Fig. 6. The operation can be described as follows: Suppose the circuit is in input I_1 (I_1=1) and in stable state A (y_1y_2=00). The output of ROM 1 first holds the partitioning variables of I_1 y_1y_2 at the current value (present state = next state). Theorem A states that this must be true. Note that the same principle is true for ROMs 2 and 3 where y_3y_4 and y_5y_6 respectively have present and next state values equal. In state A when I_1=1, the next state values for partitioning variables of I_2 and I_3 are y_3y_4=01 and y_5y_6=00 respectively, and the output is 11. If the input changes to I_2, then ROM 2 will be enabled and assume the state specified by the partitioning variables y_3y_4, namely 01; after ROM 2 is enabled, the circuit would assume stable state B with output 10. It is easy to see how the circuit would have assumed state F if I_3 had become 1 instead of I_2.

Theorem B: An asynchronous sequential flow table of m columns with up to 2^n stable states per column and Z output states can be implemented with m

ROMs of size $2^n \times w$, where $w = n \cdot m + Z$.

A design of Example A would require 8 ROMs, each 8 x 28. This is to be compared with a single ROM of size $2^{27} \times 28$ using the traditional design procedure [1]. An 8 x 28 sized ROM is very unrealistic but this demonstrates the fact that this procedure lends itself to address large problems, and since the problem was small, the resulting ROM requirements were small. If one were to utilize 2708 ROMs (1K x 8), then flow tables with up to 1024 stable states per column could be implemented. For example, a flow table with 1024 stable states in each of 4 columns would require 4 ROMs of size 1K x 40 + Z, where Z is the number of output states; if Z = 8, then a total of 4 x 6 = 24 2708 ROM bytes would be needed.

Hazards

The obvious problem with the circuit in Fig. 5 is that hazards are present as the input switches from one value to another. If $I_i=1$ (Input I_i is present) and the input was to change to I_j, then a race condition exists between ROM i turning off and ROM j turning on. This type of problem is easy to see with the design presented in this paper. However, this identical problem exists with all asynchronous sequential circuits and may be one of the major reasons designers generally avoid their use.

The following is an engineering solution to the hazard problem associated with the design presented here. Consider two input states I_i and I_j and associated ROMs as depicted in Fig. 7. Initially ignore modules E and P by assuming P is an open circuit and I_i and I_j are connected directly to the respective enable input (EN). Let $I_i=1$ and $I_j=0$. The partitioning variables I_j, PVj coming from ROM i are enabled and those of ROM j are disabled (in high impedance state) and, therefore, ROM j is presented with the correct next state value for the moment when I_j will be 1. Consider the waveforms shown in Fig. 8 when I_i and I_j change state. There are two basic problems which could occur. One is for t_1 to be so large that the feedback from ROM j has time to propagate through ROM i; this would force ROM i and ROM j to present two different values at point A. The other problem arises if t_1 is so small, perhaps negative, such that ROM j cannot enable prior to ROM i disabling. There are three cases to consider.

1) $t_{co} < t_1 < t_{acc} + t_{co}$. This case will result in proper operation. Since $t_1 > t_{co}$, ROM j has time to present a valid set of outputs (since the correct address has been present for some time, it is not necessary to wait for t_{acc} units of time for a correct output). By Theorem A, at the instant ROM j enables its output, the values of PVj from ROM i and j must agree. For the value of PVj to differ from ROM i it is necessary for the feedback value PVi from ROM j to propagate through ROM i. Since $t_1 < t_{acc} + t_{co}$, this will not happen.

2) $t_1 > t_{acc} + t_{co}$. Incorrect operation will likely occur since both ROM i and ROM j will attempt to produce conflicting values at point A, the address of ROM j.

3) $t_1 < t_{co} - t_{df}$, where t_1 could be negative. Incorrect operation will likely occur since ROM j will not have sufficient time to produce a correct output before ROM i presents an incorrect address by floating its outputs.

Cases 2 and 3 are dealt with in the following solution. A signal is needed to indicate when the output from ROM j is present. This will immediately disable ROM i. Considering Fig. 7 again, an extra output D (on data line) from each ROM is defined such that when the ROM is enabled, D will present a logic 0 (low impedance) output. P is a circuit such that whenever a transition from the high impedance state to a low impedance state occurs, a 0 to 1 to 0 pulse is generated. This pulse can be used to signal the completion of the input signal propagation through the newly enabled ROM. Circuits can be designed to produce pulses of various widths. In Fig. 9 is an example circuit of four 2-input NOR gates (a single IC chip, 74SN33) that will generate a pulse of width three gate delays. If open collector type gates are used, then a single line T can be used in Fig. 7.

Now a circuit is needed to recognize the disabling pulse. Notice, however, ROM j will see the pulse resulting from its own enabling. It will, of course, have to ignore this. Module E, the enable circuit of Fig. 7, can be implemented as an asynchronous sequential circuit to remove the problems associated with cases 2 and 3. This circuit must have the following specifications:

A) There are two inputs, I_j and T, and one output EN, the enable for ROM I_j.

B) Whenever I_j goes from 0 to 1, EN goes high to enable ROM I_j.

C) Since module P will generate a pulse, the first pulse on T after I_j goes high is ignored and EN remains high, since the pulse is from ROM I_j.

D) The second pulse on T would force EN low, disabling ROM I_j (meaning some other ROM has been enabled).

E) If I_j goes low prior to receiving a second pulse on T, EN remains high until the second pulse on T.

Specifications D and E address cases 2 and 3 respectively. Shown in Fig. 10 is a reduced flow table that specifies the above word statement, where I and T are inputs, and E is the output. Classical logic design methods can be used to design the circuit of Fig. 10, which is the worst case relative to timing. Assuming I_i is turning off, if $I_i=0$ before the leading edge of the pulse on T, then the clocked D flip flop with a SET input of Fig. 11 would implement the E-module.

Each pair of modules P and E would be the same for each input state. Furthermore, these modules would be identical for all flow tables, large or small. Each pair of modules would require between 2 and 5 ICs which is a small overhead price to be able to implement flow tables with thousands of rows. All hazard conditions are eliminated with this design and multiple input changes are permitted. The programmed ROMs and the role played by the partitioning variables of each input state eliminate the static, dynamic and function hazards. The E-module eliminates the essential hazards.

Fault Detection

The problem of real time fault detection is one of designing a circuit to detect the presence of internal faults, sometimes referred to as self-checking. Whenever a fault condition is present, a signal is generated to indicate this condition. The class of faults considered in this paper are those which affect a single chip. The fault may be a simple stuck-at-fault where one signal fails to a constant value, or a multiple fault where many signals on a chip assume a false value. It is assumed that no more than a single IC component fails simultaneously.

There are two basic approaches to detecting faults in ROM based circuits that are presented here. The first is used in a self-checking microprocessor design [8] that adds one more byte to the ROM. If there are n bits in each ROM byte and m ROM bytes, then each bit b_i, i=1,2,...,n in the m + 1 ROM byte will be the mod 2 sum of bit b_i of the m b_i bits of the other ROMs. Implementing this approach requires adding 1 byte to each ROM associated with the input states and adding external parity checkers to determine the presence of the fault. Consider the following example to demonstrate this concept:

Example B: A 4 column flow table with 4 output states requires 512 x 40 ROM for each of the input states in a non-redundant design; there are 9 partitioning variables per column. Assume 4 bit byte ROMs are used.

To add fault detection to the above non-redundant design would require adding one more byte to each ROM of the four inputs, each ROM would become 512 x 44. Four sets of parity checkers must be added; each parity checker would have 4 sets of inputs, producing 4 outputs. The external checker requires a significant amount of logic.

The second method reduces the size of the external checkers by allowing each ROM to detect the presence of faults on the address presented to the ROM. An error detecting code is generated over the partitioning variables that serve as addresses on each ROM. This must be done with care since the potential fault condition can affect all the bits on a ROM byte. A simple parity code over each partitioning variable set can be used if no more than one partitioning variable from a given set is on a single ROM byte. In example B there are 4 partitioning variable sets, each with 9 variables; each set of 9 variables must be encoded on 9 ROM bytes. With this case, one partitioning variable from each set can be placed on each of the 9 ROM bytes as shown in Fig. 12; y_j^i is defined with i = partitioning variable set (1,2,3,4) and j being the variable number in the partitioning set (1,2,3...,9). ROM byte 10 holds the parity bits where

$$P^i = \oplus \sum_{j=1}^{9} y_j^i, \quad i = 1,2,3,4.$$

In the non-redundant design, each ROM was presented with a 9 bit address. In the fault detecting design, a 10 bit (original 9 + parity) address is presented; this doubles the size of the ROMs to 1K in depth. If even (odd) parity is established over the address field, the odd (even) parity address locations are encoded to yield a fault signal. Therefore, if any single ROM byte fails, an odd (even parity location will be entered producing a fault detection signal.

In general, to detect faults in the output, the procedure outlined in the first method can be used where one more byte of ROM is added to generate parity over the output state variables. In this example with only 4 outputs, a duplicate ROM would generate the parity. The external checker would be very simple because the number of inputs is only 8; in the first method this number was 44. To detect fault conditions in real time, the size of each ROM must be 1K x 52.

40 - partition sets with parity (4 x 10)
8 - output with parity
1 - fault detection on address of ROM
3 - wasted
52

The trade off between the two approaches presented above is one of the size of the external checker and the size of the ROMs.

For fault detection to be complete, it is necessary to consider the

E and P modules. The E module can be implemented with fault detecting designs that exist [9] and could be implemented complete with real-time fault detection with approximately 5 ICs. Module P is classified as a non-normal circuit and fault detection designs do not exist as of this writing to implement that circuit; therefore, this part must be considered as "hardcore."

REFERENCES

[1]. H.A. Sholl and S.C. Yang, "Design of Asynchronous Sequential Networks Using Read-Only Memories," IEETC, pp. 195-206, Feb. 1975.

[2]. D.K. Pradham, "Fault Tolerant Asynchronous Networks Using Read-Only Memories," IEETC, pp. 674-679, July, 1978.

[3]. D. Dietmeyer, Logic Design of Digital Systems, Allyn and Bacon, 1971.

[4]. C.N. Liu, "A State Variable Assignment Method for Asynchronous Sequential Switching Circuits," J.ACM, Vol. 10 pp. 209-216, 1963.

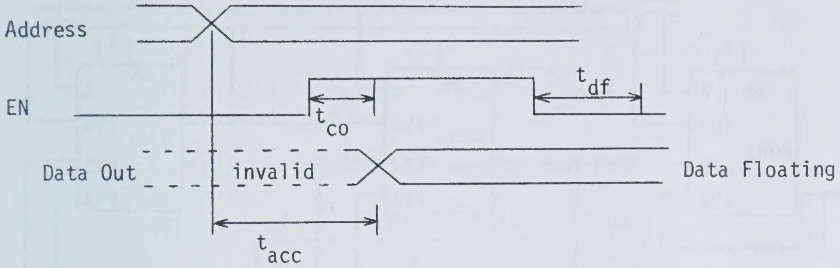
[5]. J.H. Tracey, "Internal State Assignments for Asynchronous Sequential Machines," IEETEC, pp. 551-560, Aug., 1966.

[6]. D.K. Pradham and S.M. Reddy, "Fault Tolerant Asynchronous Networks," IEETC, pp. 662-669 July, 1973.

[7]. D.H. Sawin and G.K. Maki, "Design Equation Generation for Asynchronous Sequential Circuits Using Partition Theory," Twelfth Annual Allerton Conference on Circuit and System Theory, Oct. 1974.

[8]. G.K. Maki, "A Self-Checking Microprocessor Design," Journal of Design Automation and Fault-Tolerant Computing, pp. 15-27, April, 1978.

[9]. G.K. Maki and D.H. Sawin, "Fault Tolerant Asynchronous Sequential Machines," IEETC, pp. 651-657, July, 1974.

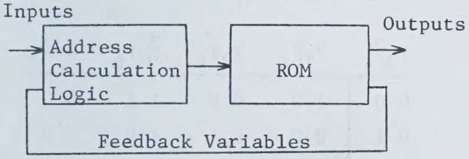


Address
EN
t_{df}
t_{co}
Data Out
invalid
Data Floating
t_{acc}

t_{acc} = Address to output delay t_{co} = Chip Select to output delay

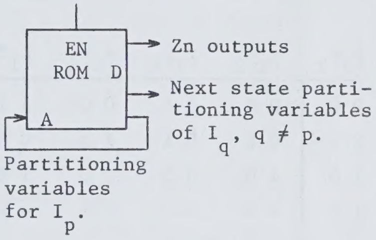
t_{df} = Chip de-select to output float

Fig. 1. Waveforms associated with a ROM.



Inputs
Address Calculation Logic
ROM
Outputs
Feedback Variables

Fig. 2. Classical approach to ROM based sequential circuits.



EN
ROM
D
A
Zn outputs
Next state partitioning variables of I_q, $q \neq p$.
Partitioning variables for I_p.

Fig. 3. ROM implementation of one column of a flow table I_p.

y_1	y_2	y_3	y_4	y_5	y_6		I_1	I_2	I_3
0	0	0	1	0	0	A	Ⓐ/11	B	F
0	0	0	1	1	0	B	A	Ⓑ/10	Ⓑ/11
1	0	0	0	1	0	C	Ⓒ/10	D	B
1	0	0	0	1	1	D	C	Ⓓ/00	E
0	1	0	1	1	1	E	Ⓔ/00	B	Ⓔ/01
0	1	1	1	0	0	F	E	Ⓕ/11	Ⓕ/00

Fig. 4. Example flow table.

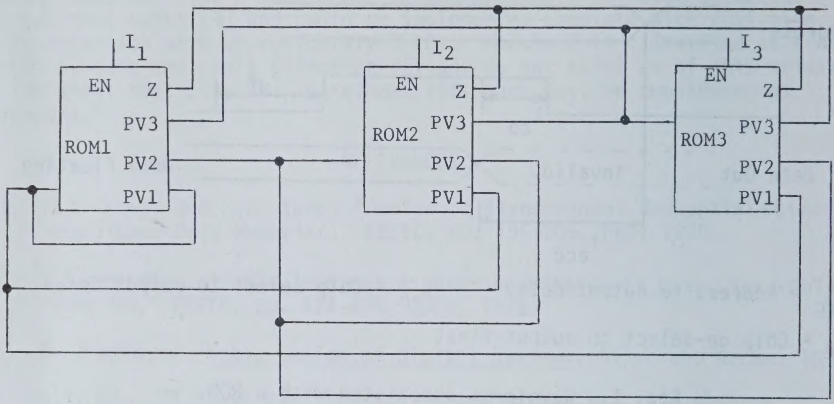

I_1
EN
Z
PV3
ROM1
PV2
PV1
I_2
EN
Z
ROM2
PV3
PV2
PV1
I_3
EN
Z
PV3
ROM3
PV2
PV1

Fig. 5. ROM design of Fig. 4.

y_1y_2	y_1y_2	y_3y_4	y_5y_6	Z_1Z_2
0 0	0 0	0 1	0 0	1 1
0 1	0 1	0 1	1 1	0 0
1 0	1 0	0 0	1 0	1 0
1 1	- -	- -	- -	- -

ROM 1

y_3y_4	y_1y_2	y_3y_4	y_5y_6	Z_1Z_2
0 0	1 0	0 0	1 1	0 0
0 1	0 0	0 1	1 0	1 0
1 0	- -	- -	- -	- -
1 1	0 1	1 1	0 0	1 1

ROM 2

y_5y_6	y_1y_2	y_3y_4	y_5y_6	Z_1Z_2
0 0	0 1	1 1	0 0	0 0
0 1	- -	- -	- -	- -
1 0	0 0	0 1	1 0	1 1
1 1	0 1	0 1	1 1	0 1

ROM 3

Fig. 6. Coded ROMs for Fig. 5.

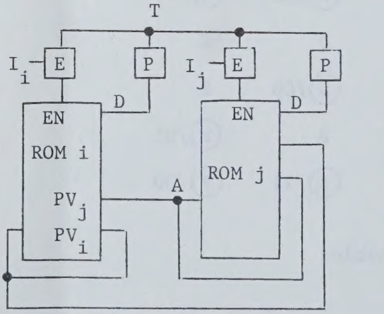

T
I_i
E
P
I_j
E
P
D
EN
ROM i
PV_j
PV_i
A
D
EN
ROM j

Fig. 7. Elimination of Hazards.

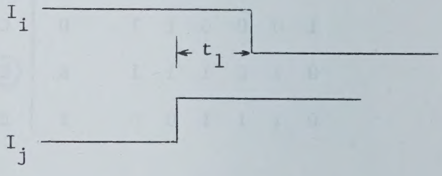


Fig. 8. Switching inputs I_i and I_j.

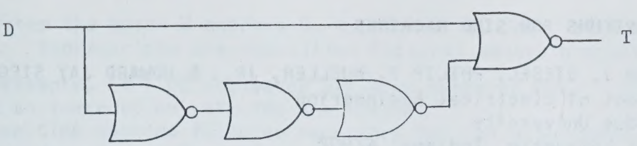


Fig. 9. Example of P module, one 74SN33.

IT	00	01	11	10
A	Ⓐ/0	D	B	Ⓐ/1
B	-	-	Ⓑ/1	C
C	Ⓒ/1	D	D	Ⓒ/1
D	A	Ⓓ/0	Ⓓ/0	Ⓓ/0

Fig. 10 Flow Table for Module E.

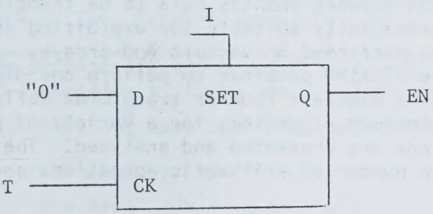

I
"0"
D
SET
Q
EN
T
CK

Fig. 11. Simplified E module.

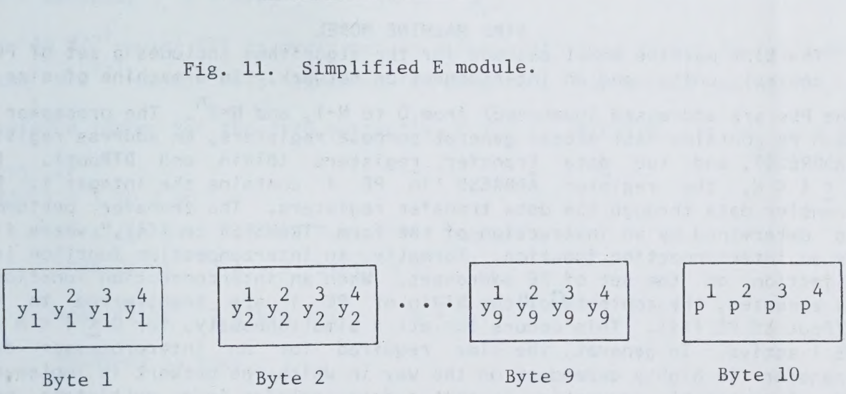

$y_1^1\ y_1^2\ y_1^3\ y_1^4$
Byte 1
$y_2^1\ y_2^2\ y_2^3\ y_2^4$
Byte 2
. . .
$y_9^1\ y_9^2\ y_9^3\ y_9^4$
Byte 9
$p^1\ p^2\ p^3\ p^4$
Byte 10

Fig. 12. Assignment of partition sets for fault detection.

FFT ALGORITHMS FOR SIMD MACHINES

LEAH J. SIEGEL, PHILIP T. MUELLER, JR., & HOWARD JAY SIEGEL
School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

ABSTRACT

Efficient computation of one- and two-dimensional discrete Fourier transforms using the SIMD (single instruction stream-multiple data stream) mode of parallel processing is demonstrated. Algorithms for a variety of problem size - machine size combinations are presented and analyzed. Radix two and radix four FFT schemes for SIMD systems are formulated.

INTRODUCTION

With the advent of microprocessors, large-scale parallel processing systems with 1024 to 4096 processors have been proposed [8,15]. Machines with 64 and 256 processors currently exist [2,3]. SIMD (single instruction stream-multiple data stream) machines [4] represent one type of parallel processing system. An SIMD machine typically consists of a control unit, a set of N processing elements (PEs), each a processor with its own memory, and an interconnection network. The control unit broadcasts each instruction to all PEs, and each active PE executes the instruction on the data in its own memory. The instruction is executed simultaneously in all active PEs. The interconnection network enables data to be transferred among the PEs. SIMD machines are especially suitable for exploiting the parallelism inherent in certain tasks performed on vectors and arrays.

In this paper, the use of SIMD machines to perform one-dimensional (1-D) and two-dimensional (2-D) discrete Fourier transforms (DFTs) [7,9] is explored. Fast Fourier transform algorithms for a variety of problem size - machine size combinations are presented and analyzed. The algorithms are evaluated in terms of the number of arithmetic operations and interprocessor transfers required.

SIMD MACHINE MODEL

The SIMD machine model assumed for the algorithms includes a set of PEs, a control unit, and an interconnection network. In a machine of size N, the PEs are addressed (numbered) from 0 to N-1, and $N=2^n$. The processor in each PE contains fast access general purpose registers, an address register (ADDRESS), and two data transfer registers (DTRin and DTRout). For $0 \leq i < N$, the register ADDRESS in PE i contains the integer i. PEs transfer data through the data transfer registers. The transfer performed is determined by an instruction of the form "TRANSFER to f(i)," where f(i) is an interconnection function. Formally, an interconnection function is a bijection on the set of PE addresses. When an interconnection function f is executed, the contents of the DTRin of PE i are transferred to the DTRout of PE f(i). This occurs for all i simultaneously, for $0 \leq i < N$ and PE i active. In general, the time required for an interprocessor data transfer is highly dependent on the way in which the network is implemented. A reasonable assumption is that a data transfer in a multistage network requires no more time than a complex multiplication.

PE address masks [10] are used to enable and disable PEs. In this scheme, an n-position mask specifies which of the N PEs are to be activated for each instruction. Each mask position contains a 0,1, or X ("don't care") and the only PEs active are those whose address (in binary represen-

---

This research was supported by NSF Grant No. ENG-78-05789 and by the U.S. Air Force, Rome Air Development Center, under Contract No. F30602-78-C-0025.

tation) matches the mask: 0 matches 0, 1 matches 1, and either 0 or 1 matches X. Superscripts are repetition factors; square brackets denote a mask. For example, $[X^{n-1}0]$ activates all even numbered PEs. If no mask accompanies an instruction, all PEs are active.

Only those SIMD machine features necessary for the algorithms that follow have been described. The model presented is intended to provide a very general framework in which SIMD algorithms can be developed.

ONE-DIMENSIONAL FFT ALGORITHMS

The algorithm in Figure 1 computes the M-point DFT of the signal $s(m)$, $0 \leq m < M$, using a machine having $N=M/2$ PEs [13]. The algorithm is a parallel implementation of a radix two decimation-in-frequency technique [7,9]. The decimation-in-frequency algorithm divides the input sequence $\{s(m)\}$ into sequences $\{s_1(m')\}$, equal to the first half of $\{s(m)\}$, and $\{s_2(m')\}$, equal to the second half of $\{s(m)\}$. The DFT of the M-point sequence can then be computed by calculating the two M/2-point DFTS of the sequences

$$\{s_1(m') + s_2(m')\} \text{ and } \{[s_1(m') - s_2(m')]\ W_M^{m'}\}$$

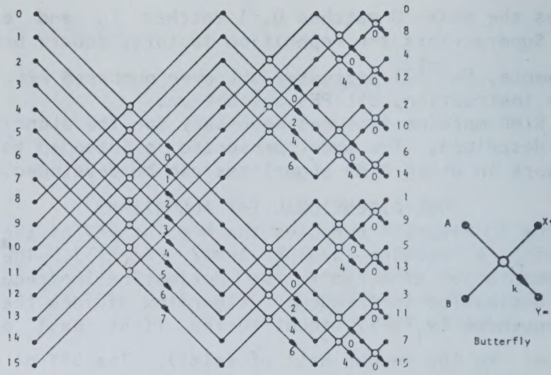
where $0 \leq m' < M/2$ and $W_M = e^{-2\pi j/M}$. Figure 2 shows the computations performed in a radix two 16-point FFT.

For the parallel algorithm, PE i initially contains $s(i)$ and $s(i + M/2)$, $0 \leq i < N$. The data transfers are specified in terms of the Cube interconnection network. The network consists of n interconnections, Cube_i,

---

```
/* Compute the M-point FFT using PEs 0 to M/2-1, where initially the A register of PE i contains s(i) and the B register of PE i contains s(i+M/2). The "twiddle factors" $W_M^k$ have been precomputed in each PE. $z = \log_2 M-1$. */
/* butterfly for M-point FFT */
$k \leftarrow$ ADDRESS;   $X \leftarrow A+B$;   $Y \leftarrow (A-B) * W_M^k$;
/* An $M/2^i$-point FFT is performed by a butterfly (see Figure 2) followed by two $M/2^{i+1}$-point FFTs executed in parallel. For an $M/2^i$-point FFT, the twiddle factor for the r-th butterfly, $0 \leq r < (M/2^i)/2$, is $W^r_{M/2^i} = W_M^{r2^i}$ [9]. */
for i ← 1 to z do
    /* align data for $M/2^i$-point FFT butterfly */
    MASK $[X^i 0 X^{z-i}]$ $A \leftarrow X$; $DTRin \leftarrow Y$;
    MASK $[X^i 1 X^{z-i}]$ $DTRin \leftarrow X$;   $B \leftarrow Y$;
    TRANSFER to $\text{Cube}_{z-i}$;
    MASK $[X^i 0 X^{z-i}]$ $B \leftarrow DTRout$;
    MASK $[X^i 1 X^{z-i}]$ $A \leftarrow DTRout$;
    /* perform $2^i$ $M/2^i$-point FFT butterflies in parallel */
    $k \leftarrow (\text{ADDRESS} * 2^i) \bmod M/2$;   $X \leftarrow A+B$;   $Y \leftarrow (A-B) * W_M^k$;
```

Figure 1: Algorithm to compute the M-point FFT in M/2 PEs.



0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
0
8
4
12
2
10
6
14
1
9
5
13
3
11
7
15
A
B
X=A+B
Y=(A-B)W_M^k
k
Butterfly

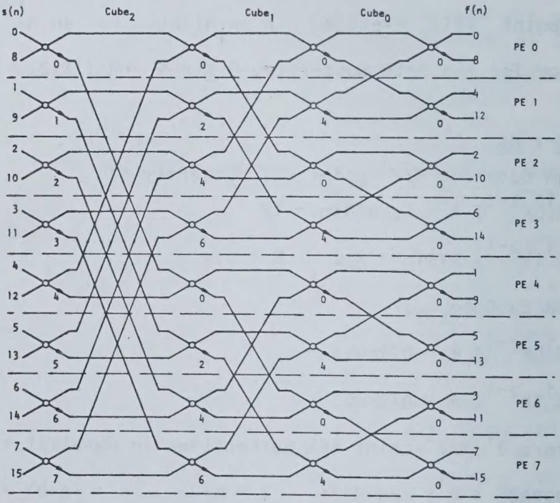
Figure 2: Operations in the computation of a 16-point FFT.

$0 \leq i < n$, defined as

$$\text{Cube}_i\,(p_{n-1}\cdots p_{i+1}p_ip_{i-1}\cdots p_0) = p_{n-1}\cdots p_{i+1}\overline{p}_ip_{i-1}\cdots p_0\,,$$

where $p_{n-1}\cdots p_0$ is the binary representation of an arbitrary PE address, and $\overline{p}_i$ is the complement of p_i. The Cube interconnections are realizable by a number of networks discussed in the literature, including PASM's ADM network [11,13], Pease's indirect binary n-cube [8], Lawrie's omega network [6], and the STARAN flip network [2]. Figure 3 illustrates the pattern of data transfers and computations performed in the parallel algorithm.

The algorithm performs the M-point FFT calculations using $\log_2(M/2)$ parallel data transfers. The number of parallel butterfly operations performed is $\log_2 M$ where each butterfly involves two complex additions and one complex multiplication. The reduction in the number of butterfly steps is



s(n)
Cube$_2$
Cube$_1$
Cube$_0$
f(n)
PE 0
PE 1
PE 2
PE 3
PE 4
PE 5
PE 6
PE 7

Figure 3: Computation of a 16-point FFT in 8 PEs.

from $(M/2)\log_2 M$ for a serial algorithm to $\log_2 M$ for the parallel algorithm. This reduction by a factor of M/2 is the maximum achievable with M/2 PEs.

A lower bound on the number of data transfer steps required to perform an M-point FFT when the M points are initially distributed over M/2 PEs can be obtained using a measure of data movement developed by Gentleman [5]. The function sigma(k) is defined to be the maximum number of processors at which data originally available in a single processor can be made available in k or fewer data movement steps [5].

Theorem 1: Computation of an M-point FFT on an SIMD machine with M/2 PEs interconnected by a Cube network, where the M data points are distributed evenly over the M/2 PEs, requires at least $\log_2(M/2)$ data transfer steps.

Proof: By definition of the DFT, each element in the FFT sequence {f(m)} depends on every element of the input sequence {s(m)}. Therefore there must be a path from any PE containing a f(m) (i.e. PEs 0 to M/2-1) to each of the PEs containing s(m)'s. Let d be the number of Cube functions needed for these transfers. Since each PE must receive data from M/2 other PEs, $\text{sigma}(d) \geq M/2$. For the Cube network, $\text{sigma}(k) = 2^k$ (consider performing k different Cube_i functions). Therefore $d \geq \log_2(M/2)$. []

Because M/2 PEs may not be available for the computation of an M-point DFT, it is of interest to consider FFT algorithms which use fewer PEs. As an example of possible techniques, algorithms using N=M/4 PEs to compute the M-point transform are discussed.

A simple solution to the problem of having fewer PEs is to replicate the steps in the radix two, M/2 PE algorithm. Two computations that were performed in parallel in different PEs in the M/2 PE algorithm are now performed sequentially in the same PE, as shown in Figure 4. The number of radix two butterfly steps performed is $2\log_2 M$, so $4\log_2 M$ complex addition and $2\log_2 M$ complex multiplication steps are executed. $2(\log_2 M-1)-2$ parallel data transfers are required.

An alternative approach when M is a power of four performs the operations of a radix four decimation-in-frequency FFT. The input sequence {s(m)} is divided into four subsequences $\{s_i(m')\}$ of M/4 points, consisting of the s(m) for, respectively, $0 \leq m < M/4$, $M/4 \leq m < M/2$, $M/2 \leq m < 3M/4$,

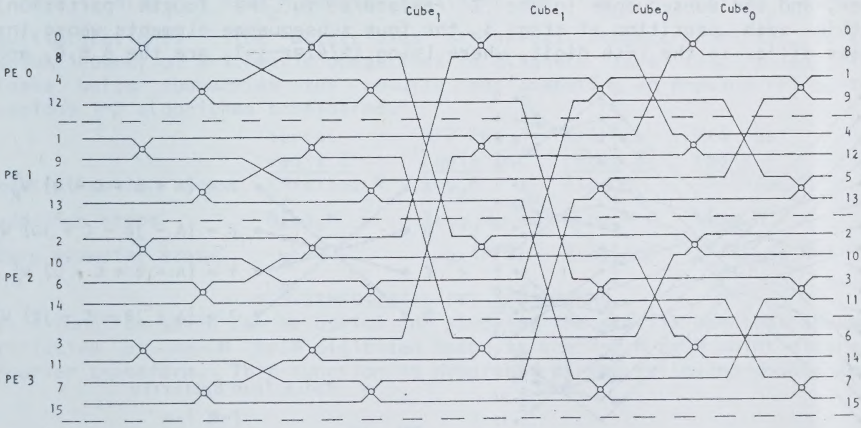


Figure 4: Computation of a 16-point FFT in 4 PEs, using a radix two algorithm.

and $3M/4 \le m < M-1$. The DFT of the M-point sequence can be computed by calculating the four M/4-point DFTs of the sequences

$$\{s_1(m')+s_2(m')+s_3(m')+s_4(m')\} \tag{1}$$

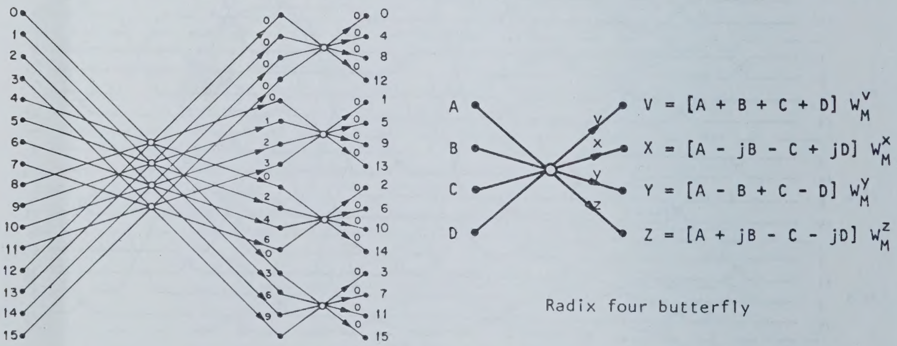
$$\{[s_1(m')-js_2(m')-s_3(m')+js_4(m')]W_M^{m'}\} \tag{2}$$

$$\{[s_1(m')-s_2(m')+s_3(m')-s_4(m')]W_M^{2m'}\} \tag{3}$$

$$\{[s_1(m')+js_2(m')-s_3(m')-js_4(m')]W_M^{3m'}\} \tag{4}$$

where $0 \le m' < M/4$ and $W_M = e^{-2\pi j/M}$. Figure 5 shows the computations performed in a radix four 16-point FFT. Let $r = \log_4 M$, and let $m = d_{r-1}\ldots d_1 d_0$ be the base four representation of m, $0 \le m < M$, $0 \le d_i < 4$. At stage i in the computation, where the stages are numbered from 0 to r-1, the four items whose indices differ in the digit d_{r-i-1} are combined in the radix four butterfly operation.

An algorithm to compute the M-point DFT using N=M/4 PEs will consist of r stages of four-input butterflies, with the interconnection network used to align the data at each stage. PE j initially contains the four points s(j), s(j+M/4), s(j+M/2), and s(j+3M/4), $0 \le j < M/4$. Using the butterfly notation of Figure 5, registers A, B, C, and D of PE j are initially loaded with points s(j), s(j+M/4), s(j+M/2), and s(j+3M/4) respectively, $0 \le j < M/4$. The result of one butterfly executed in parallel in all PEs is to generate the four subsequences of equations (1)-(4), one in each of the sets of V, W, Y, and Z registers. From this point, the same processing steps are performed for each stage of the FFT, and for each subsequence. It therefore suffices to consider the actions performed in one partition consisting of $Q=M/4^i$ PEs. The butterfly operation in stage i-1 of the FFT, $0 < i < r$, generated the four subsequences of equations (1)-(4), with one $Q=M/4^i$-point subsequence in each of the V,X,Y, and Z registers of the partition. In order to process each Q-point subsequence, the size-Q partition is quartered. The subsequence in the V registers will be transferred to the first Q/4-size partition, the subsequence in the W registers to the second partition, the subsequence in the Y registers to the third partition, and the subsequence in the Z registers to the fourth partition. Within each partition at stage i, the four subsequence elements whose indices differ in the l-th digit, where $l=\log_4(Q/4)=r-i-1$, are the A,B,C, and


0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
0
0
0
0
0
1
2
3
0
2
4
6
0
3
6
9
0
4
8
12
1
5
9
13
2
6
10
14
3
7
11
15
A
B
C
D
v
x
y
z
$V = [A + B + C + D]\,W_M^v$
$X = [A - jB - C + jD]\,W_M^x$
$Y = [A - B + C - D]\,W_M^y$
$Z = [A + jB - C - jD]\,W_M^z$

Radix four butterfly

Figure 5: Operations in the computation of radix four 16-point FFT [9].

D inputs to a butterfly. The data transfers from the size-Q partition to the four size-Q/4 partitions must therefore be done so that these four elements are in the same PE. The table below summarizes the transfers of the data in the V,X,Y, and Z registers that must be performed. Arithmetic is mod Q, so that the PEs in each size-Q partition are considered to be numbered from 0 to Q-1; Q'=Q/4 is the size of the new partitions. An entry of the form "+k" for PE j means that the specified element is to be moved from PE j to PE (j+k) mod Q. An entry of the form "-" means that no inter-PE transfer is needed.

PE #j	V	X	Y	Z	DESTINATION
$0 \leq j < Q'$	----	+Q'	+2Q'	+3Q'	A register
$Q' \leq j < 2Q'$	+3Q'	----	+Q'	+2Q'	B register
$2Q' \leq j < 3Q'$	+2Q'	+3Q'	----	+Q'	C register
$3Q' \leq j < Q$	+Q'	+2Q'	+3Q'	----	D register

The transfers needed at each stage can be accomplished in three parallel data transfers if the interconnection network can perform the uniform shifts mod Q in a single step. An algorithm using the Cube functions to do the +Q' mod Q transfer for PEs 4^k to 4^k+Q-1 is:

```
for j = 2k to 2k+$\log_2 Q-1$ do     MASK [$X^{n-j}\ 0^{j-2k}\ X^{2k}$] $\mathrm{Cube}_j$
```

The +2Q' mod Q transfer is similar. The +3Q' mod Q = -Q' mod Q transfer can be done using Cube functions as follows:

```
for j = 2k to 2k+$\log_2 Q-1$ do     MASK [$X^{n-j}\ 1^{j-2k}\ X^{2k}$] $\mathrm{Cube}_j$
```

These algorithms, based upon the "Cube → PM2I" algorithm in [12], can be implemented in one pass through Pease's indirect binary n-cube [8] and Lawrie's omega network [6]. (The relationship between Cube functions and these networks is presented in [11].) The ADM network can do any transfers the Cube network can [14].

The total number of data transfer steps for the radix four algorithm to compute an M-point DFT in M/4 PEs is $3(\log_4 M-1) = (3/2)\log_2 M-3$. The four-input butterfly which generates the sequences of equations (1)-(4) can be computed in eight complex additions and three nontrivial complex multiplications. (There is also one trivial multiplication by j.) There are $\log_4 M$ butterfly steps, so the total number of arithmetic operation steps performed is $8\log_4 M = 4\log_2 M$ complex additions and $3\log_4 M = (3/2)\log_2 M$ complex multiplications. (The butterflies at the last stage are trivial, so the actual number of arithmetic operations is slightly less than this.) The table below summarizes the results for computing an M-point FFT by the various 1-D algorithms considered.

	serial radix 2	M/2 PEs radix 2	M/4 PEs radix 2	M/4 PEs radix 4
multiplication steps	$(M/2)\log_2 M$	$\log_2 M$	$2\log_2 M$	$(3/2)\log_2 M$
addition steps	$M\log_2 M$	$2\log_2 M$	$4\log_2 M$	$4\log_2 M$
data transfer steps	----	$\log_2(M/2)$	$2\log_2 M-4$	$(3/2)\log_2 M-3$

TWO-DIMENSIONAL FFT ALGORITHMS

A function which can be useful for studying the spacial spectral characteristics of an M by M digitized image is the 2-D M by M point discrete Fourier transform. This function is described by the following double sum:

$$F(v,w) = \sum_{l=0}^{M-1} \sum_{m=0}^{M-1} S(l,m)\, W_M^{vl}\, W_M^{wm} \qquad 0 \leq v,w < M$$

where S is the input image, and F is the DFT of S.

A standard approach to evaluating this function efficiently is to rewrite the double sum as 2M 1-D DFTs [9]:

$$G(l,w) = \sum_{m=0}^{M-1} S(l,m)\, W_M^{wm} \qquad 0 \le l,w < M, \tag{5}$$

$$F(v,w) = \sum_{l=0}^{M-1} G(l,w)\, W_M^{vl} \qquad 0 \le v,w < M. \tag{6}$$

G is the 1-D DFT of the rows of S and F is the 1-D DFT of the columns of G. These 2M 1-D DFTs are usually implemented using FFT techniques.

Suppose that an SIMD machine has N=M PEs, each of which has one row of an M by M input image, S. An efficient method for obtaining F, the DFT of S, is to perform M 1-D FFTs in parallel on the rows of S to get G, then perform M 1-D FFTs in parallel on the columns of G to get F. The first step is to perform 1-D DFTs on the M rows of S simultaneously, one DFT in each processor, using a serial FFT algorithm. The result, G, is given by (5) above. This step achieves an ideal speedup of a factor of M. The second step is to form the transpose of G, G', such that each row of G' is in a different PE. In general, G(i,j) is sent from PE i to PE j. This is done by having G(i,i+k mod M) sent from PE i to PE i+k mod M, 0<k<M. For each value of k, this type of uniform shift can be done in one pass through networks such as the n-cube, omega, and ADM [8,6,14]. This step requires M-1 parallel data transfers for these networks. Figure 6 gives an algorithm to perform the transpose operation. The final step is to perform 1-D DFTs on the rows of G' (the columns of G) to obtain F' as given by (6) above. These M 1-D DFTs are performed simultaneously, one in each PE, using a serial FFT algorithm. The result, F', can be transposed to give F, however, this may not be necessary depending on what further processing is done on F. Again a perfect speedup of M is achieved.

The serial complexity of a 1-D FFT is $(M/2)\log_2 M$ radix two butterflies. The above parallel implementation of the 2-D DFT executes two serial FFT algorithms and has a complexity of $M\log_2 M$ butterfly steps. The serial complexity of 2M 1-D FFTs (i.e., an M by M 2-D DFT) is $M^2\log_2 M$ butterflies. Thus, an ideal speedup of M is achieved for butterfly operations with a cost of M-1 data transfers.

Given assumptions about the initial data placement, a lower bound on the

---

```
/* Transpose M by M array G, placing result in G'.  PE  k  receives  G(j,k)
from  PE  j  as  follows:  At step i, G(j,k)s for which (k-j) mod M = i are
transferred from PE j to PE k. In each PE, G denotes  the  address  of  the
first element of the G vector that is stored in that PE; G' denotes the ad-
dress of the first element of the G transpose vector in each PE. b bytes of
storage are needed for each array element. */

for i ← 1 to M-1 do
    /* in PE j, send G(j,k), k = j + i */
    DTRin ← G + b * (ADDRESS + i) mod M;    TRANSFER to PE + i;
    /* in PE k, received G(j,k), j = (k-i) mod M */
    G' + b * (ADDRESS - i) mod M ← DTRout;
G' + b * ADDRESS ← G + b * ADDRESS     /* copy diagonal from G to G' */
```

Figure 6: Algorithm to transpose an M by M array using M PEs, where initially PE j holds row j of the array.

number of data transfer steps required to transpose the G array can be proven.

Theorem 2: If M by M array G is distributed over M PEs, with row i in PE i, then a lower bound on the number of data transfer steps required to transpose G so that PE i contains column i is M-1.
Proof: Element G(i,j) must be transferred from PE i to PE j, for all $0 \le i, j < M$ and $i \neq j$. The total number of elements which must be transferred is therefore M(M-1). At most M elements can be moved in one data transfer step, so at least M-1 data transfer steps are needed.

□

This approach can be generalized for $N \le M$. For example, if N=M/2, a speedup of approximately N can be achieved. For this implementation each PE is given two rows of the input matrix S. The FFTs on the rows of S are performed by two serial FFTs, executed one after the other, on the two rows in each PE. This yields G as described by (5), with each PE having two rows of G. The second step is to form the transpose of G, G', where each PE has two rows of G' (i.e., each PE has two columns of G). If PE i contains rows 2i and 2i+1, then, in general, G(i,j) is transferred from PE $\lfloor i/2 \rfloor$ to PE $\lfloor j/2 \rfloor$. The complexity associated with the transpose is 2M-4 parallel transfers. The -4 term appears because the diagonal and near diagonal terms are already in the correct PE. The final step is to perform a 1-D DFT on the columns of G. This is done by two serial FFTs in each PE, as above. This gives F, with each PE having two columns of F. This implementation has a complexity of four serial FFT algorithms, or $2M\log_2 M$ butterfly steps. This is the maximum possible reduction in the number of butterflies. The overhead associated with the transpose is 2M-4 transfers.

In the case where $N=M^2/4$, a more efficient approach can be used. Instead of reducing the 2-D DFT to 1-D DFTs, a 2-D decimation-in-frequency algorithm is derived. The 2-D FFT is obtained by dividing the matrix into four quadrants and obtaining expressions for the M by M point DFT in terms of four M/2 by M/2 point DFTS. The four DFTs are given by:

$$F_i(v,w) = \sum_{l=0}^{M/2-1} \sum_{m=0}^{M/2-1} S_i(l,m)\, W_{M/2}^{lv/2}\, W_{M/2}^{mw/2} \qquad 0 \le v,w < M$$

for $0 \le i < 4$, where for F_0 v and w are even, for F_1 v is even and w is odd, for F_2 v is odd and w is even, for F_3 v and w are odd. The S_i's are given by

$$S_0(l,m) = S(l,m) + S(l,m+M/2) + S(l+M/2,m) + S(l+M/2,m+M/2)$$

$$S_1(l,m) = [S(l,m) - S(l,m+M/2) + S(l+M/2,m) - S(l+M/2,m+M/2)]\, W_M^m$$

$$S_2(l,m) = [S(l,m) + S(l,m+M/2) - S(l+M/2,m) - S(l+M/2,m+M/2)]\, W_M^l$$

$$S_3(l,m) = [S(l,m) - S(l,m+M/2) - S(l+M/2,m) + S(l+M/2,m+M/2)]\, W_M^m W_M^l$$

for $0 \le l,m < M/2$. This process is repeatedly applied to each 2-D DFT to obtain a 2-D decimation-in-frequency algorithm.

The SIMD machine is viewed as an M/2 by M/2 matrix of PEs. Each PE is given a logical address of the form (i,j), with logical PE (i,j) corresponding to physical PE iM/2+j. PE (i,j) is given S(i,j), S(i,j+M/2), S(i+M/2,j), and S(i+M/2,j+M/2). This algorithm consists of $\log_2 M$ steps. Each step can be divided into two parts: the parallel data transfers to align the data, and the computations required to compute the expressions

S_0, S_1, S_2, and S_3.

No parallel transfers are required in the first step since $S(l,m)$, $S(l,m+M/2)$, $S(l+M/2,m)$, and $S(l+M/2,m+M/2)$ are together in PE (l,m), $0 \leq l, m < M/2$. Thus, the data is aligned and S_0, S_1, S_2, and S_3 are computed. At the end of the first step, PE (l,m) contains one point from each of the four DFTs which need to be computed to give the required M by M DFT. The next step is to divide the set of PEs, PE (l,m), $0 \leq l,m < M/2$, into four quadrants. Each quadrant will compute one of the four DFTs. The data transfers for this step must put the data for each DFT into a different quadrant. Consider the case where there are N PEs and square root of N is an integer. If the array of PEs is partitioned into quadrants, numbered in clockwise order with the upper left being 0, then: (1) PEs in quadrants 0 and 1 and the corresponding PEs in quadrants 3 and 2 can exchange data using the Cube_{n-1} interconnection function; (2) PEs in quadrants 0 and 3 and quadrants 1 and 2 can exchange data using the $\text{Cube}_{n/2-1}$ function; and (3) PEs in diagonally opposite quadrants can exchange data using the $\text{Cube}_{n/2-1}$ and Cube_{n-1} functions. Any of the networks discussed previously can do each of these exchanges in one transfer [11]. After the data is aligned, S_0, S_1, S_2, and S_3 are computed for each of the four DFTs.

This process of dividing each set of PEs into four quadrants, and then assigning a different DFT to each quadrant is repeated. Interconnection functions analogous to those described above can be used to align the data. The total number of steps (including the first step) is $\log_2 M$. After all $\log_2 M$ steps have been executed, each PE contains four values, each of which is an element in the M by M transform matrix.

The computation of S_0, S_1, S_2, and S_3 at each step requires four complex multiplications and eight complex additions. Three parallel data transfers are needed to align the data at each step after the first step. Thus, the total complexity is $3(\log_2 M-1)$ transfers plus $4\log_2 M$ multiplications and $8\log_2 M$ complex additions. The serial complexity is $M^2\log_2 M$ multiplications and $2M^2\log_2 M$ additions, so the speedup for arithmetic operations is $M^2/4$.

CONCLUSIONS

Parallel 1- and 2-D FFT algorithms for a variety of problem size-machine size combinations have been presented and analyzed. The design and analysis of parallel algorithms differs from that of serial algorithms in that the asymptotic time complexity of a serial algorithm will be approximately the same for any canonic model of a serial computer [1]. As has been shown by the algorithms above, the complexity of parallel algorithms depends on architecture features such as the number of PEs and the type of interprocessor connection network. Future work includes a study of how other factors, such as the initial data allocation, affect the complexity.

REFERENCES

[1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA, 1976.

[2] K. E. Batcher, "STARAN parallel processor system hardware," *AFIPS Conf. Proc. 1974 NCC*, Vol. 43, May 1974, pp. 405-410.

[3] W. J. Bouknight, et. al., "The Illiac IV system," *Proc. IEEE*, Vol. 60, Apr. 1972, pp. 369-388.

[4] M. J. Flynn, "Very high-speed computing systems," *Proc. IEEE*, Vol. 54, Dec. 1966, pp. 1901-1909.

[5] W. M. Gentleman, "Some complexity results for matrix computations on parallel processors," JACM, Vol. 25, Jan. 1978, pp. 112-115.
[6] D. H. Lawrie, "Access and alignment of data in an array processor," IEEE Trans. Comp., Vol. C-24, Dec. 1975, pp. 1145-1155.
[7] A. V. Oppenheim and R. W. Schafer, Digital Signal Processing, Prentice-Hall, Englewood Cliffs, NJ, 1975.
[8] M. C. Pease, "The indirect binary n-cube microprocessor array," IEEE Trans. Comp., Vol. C-26, May 1977, pp. 458-473.
[9] L. R. Rabiner and B. Gold, Theory and Application of Digital Signal Processing, Prentice-Hall, Englewood Cliffs, NJ, 1975.
[10] H. J. Siegel, "Analysis techniques for SIMD machine interconnection networks and the effects of processor address masks," IEEE Trans. Comp., Vol. C-26, Feb. 1977, pp. 153-161.
[11] H. J. Siegel, "Interconnection networks for SIMD machines," Computer, Vol. 12, June 1979, pp. 57-65.
[12] H. J. Siegel, "A model of SIMD machines and a comparison of various interconnection networks," IEEE Trans. Comp., to appear Dec. 1979.
[13] H. J. Siegel, L. J. Siegel, R. J. McMillen, P. T. Mueller, Jr., and S. D. Smith, "An SIMD/MIMD multimicroprocessor system for image processing and pattern recognition," IEEE Conf. Pattern Recognition and Image Processing, Aug. 1979, pp. 214-224.
[14] H. J. Siegel and S. D. Smith, "Study of multistage SIMD interconnection networks," 5th Symp. Comp. Arch., Apr. 1978, pp. 223-229.
[15] H. Sullivan, et al., "A large scale homogeneous, fully distributed parallel machine," 4th Symp. Comp. Arch., Mar. 1977, pp. 105-124.

CAPACITY CALCULATION OF COMPOSITE CONCENTRATORS

SHINJI NAKAMURA AND GERALD M. MASSON
Department of Electrical Engineering
The Johns Hopkins University
Baltimore, Maryland 21218

ABSTRACT

In this paper, a capacity calculation technique is described for a class of interconnection networks called composite concentrators. This technique is then used to develop some capacity observations for various concentrators of this class.

INTRODUCTION

A concentrator is a type of interconnection network having switching elements, called crosspoints, between disjoint sets of inputs and outputs to the network such that specified sets of inputs of some maximum size can be interconnected to arbitrary sets of outputs of the same size [1]. More specifically, a concentrator with N inputs and M outputs, $N > M$, is said to have a capacity $c \leq M$ if, for any choice of $K \leq c$ inputs, a set of K disjoint paths from the inputs can be established by closing crosspoints to some K outputs. It is important to note that the inputs can be specified, but the outputs to which they are connected cannot be specified. Clearly, if crosspoints were placed between every input and output, the capacity would be equal to the number of outputs. (Indeed, for such a network, the outputs could also be specified.) However, in general, the number of crosspoints would be prohibitively high; hence, sparser constructions are of interest. The problem with which we deal in this paper is the capacity calculation of a class of one-stage sparse crosspoint networks called composite concentrators.

Preliminaries

Figure 1 shows three examples of one-stage (or bipartite) sparse crosspoint networks. Such networks can be expressed formally as a triplet (I,O,R), where I is the set of inputs, O is the set of outputs, and R is a relation detailing the crosspoint placement between the inputs and the outputs. R can be given in the form of a graph, or as a crosspoint model (as is done in Figure 1), or as a set of ordered pairs where $(i,j)\varepsilon R$ if there is a crosspoint between input i and output j. Regarding the latter, we can express the network of Figure 1(a) as

$$(\{1,2,3,4,5\}, \{a,b,c\}, \{(1,a),(1,c),(2,b),(3,c),(4,b),(4,c),(5,a),(5,b)\}) .$$

Similarly, the networks of Figures 1(b) and 1(c) can respectively be expressed as

$$(\{1,2,3,4,5,6\},\{a,b,c,d\},\{(1,a),(1,b),(2,a),(2,c),(3,a),(3,d),(4,b),(4,c),(5,b),(5,d),(6,c),(6,d)\})$$

and

$$(\{1,2,3\},\{a,b,c\},\{(1,a),(1,b),(2,a),(2,c),(3,b),(3,c)\}) .$$

It should be noted that the networks of Figures 1(a) and 1(b) satisfy the implicit requirement for concentrators that the number of inputs is greater than the number of outputs and (since the capacity must be greater than or equal to one and all outputs must be of some use in providing interconnecting paths from the inputs through the network) that each input (output) can potentially be connected to at least one output (input). More formally, we can express one-stage concentrators as triplets (I,O,u) where

$|I|>|O|$, and where u (u^{-1}) is a relation mapping elements of $I(O)$ to subsets of $O(I)$ corresponding to the crosspoint placement such that $u(i)\neq\phi$ $(u^{-1}(o)\neq\phi)$ for all $i\varepsilon I$ $(o\varepsilon O)$.

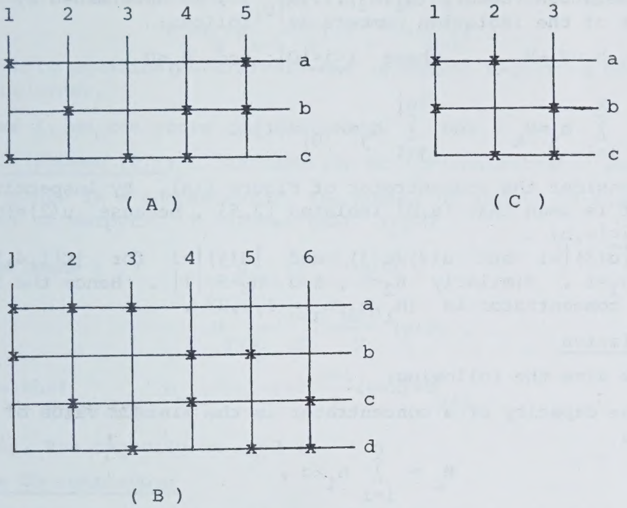


Figure 1

It should also be pointed out for later reference that Figure 1(b) is a basic binomial concentrator which we will describe as a $\binom{4}{2}$-concentrator since the crosspoint pattern connecting the outputs to the inputs consists of all possible $\binom{4}{2}$ choices of 2 crosspoints between the inputs and the 4 outputs. In general, an $\binom{x}{y}$-concentrator has $\binom{x}{y}$-inputs, x-outputs, $y\cdot\binom{x}{y}$ crosspoints. It should also be noted that the network of Figure 1(c) is $\binom{3}{2}$-network, but it is not a concentrator.

The concentrator of Figure 1(a) has a capacity of 2 and the $\binom{4}{2}$-concentrator of Figure 1(b) has a capacity of 4 . However, the capacity of a sparse crosspoint networks which satisfies the implicit requirements for a concentrator is, in general, very difficult to determine. Clearly, one way to determine the capacity is to exhaustively consider subsets of inputs (starting with subsets of size two and progressing upwards) until a subset is found for which there is not a one-to-one mapping of the inputs to some subset of the outputs. However, even for networks of modest size this is prohibitive. Hence, in this paper we will develop a much more efficient capacity calculation technique with which to address this problem for a class of concentrators called composite concentrators.

Isolation Concepts

To develop our capacity calculation scheme it is necessary that we first define the following concepts.

For a concentrator expressed as (I,O,u), if $u(i)\subseteq O'\subseteq O$ for some $i\varepsilon I$, then we will say that i is completely enclosed by O' , or O' isolates i . For a given subset $O'\subseteq O$, the set $\sigma(O')$ is:

$$\sigma(O') = \{i \mid i\varepsilon I \text{ and } u(i)\subseteq O'\} .$$

$\sigma(O')$ is called the isolated set of O' , since elements of $\sigma(O')$ cannot access any output except those which are the elements of O' .

For all subsets O' of O for which $|O'|=k$, we define an isolation number H_k to be:

$$H_k = \max_{|O'|=k \text{ and } O'\subseteq O} |\sigma(O')| .$$

Finally, the isolation vector, $(h_1,h_2,\ldots,h_{|O|})$, is determined by taking the differences of the isolation numbers as follows:

$$h_j=H_j-H_{j-1} \quad \text{where} \quad 1\leq j\leq |O| \quad \text{and} \quad H_0=0 .$$

Obviously $\quad \sum_{j=1}^{k} h_j=H_k$, and $\sum_{j=1}^{|O|} h_j=H_{|O|}=|I|$.

Example 1: Consider the concentrator of Figure 1(a). By inspecting the relation u it is seen that $\{a,b\}$ isolates $\{2,5\}$, because $u(2)=\{b\}\subset\{a,b\}$. and $u(5)=\{a,b\}\subseteq\{a,b\}$. Since $|u(2)|=|u(3)|=1$ but $u(2)\neq u(3)$ and $|u(j)|>1$ for $j\in\{1,4,5\}$, it follows that $H_1=1$. Similarly $H_2=3$, and $H_3=5=|I|$. Hence the isolation vector of this concentrator is $(h_1,h_2,h_3)=(1,2,2)$.

Capacity Calculation

We can now give the following:

Theorem 1: The capacity of a concentrator is the minimum value of c which satisfies

$$H_c = \sum_{i=1}^{c} h_i>c ,$$

where $(h_1,h_2,\ldots,h_{|O|})$ is the isolation vector of the concentrator.

Proof: If c is the minimum value which satisfies $H_c>c$, then any H_j for $1\leq j<c$ is less than or equal to j , that is , $H_j\leq j$ for $1\leq j<c$. Therefore any subset I' of I , where $|I'|=j<c$, has crosspoints to a subset $O'=\cup_{i\in I'}u(i)\leq O$ and $|O'|\geq j$. Thus from Hall's Theorem (see Appendix), any subset $I'\subseteq I$ of size up to and including $c-1$ has some subset $O'\subseteq O$ where there is a one-to-one correspondence between I' and O' . Now from the requirement on the concentrator that for any $i\in I$, $u(i)\neq\phi$, it follows that for the given subset O' , every element $i\notin\sigma(O')$ is connected at least one element in O which is not in O' ; that is,

$$\text{for all } i\notin\sigma(O') , \quad u(i)\cap(O-O')\neq\phi .$$

Therefore for any subset $I'\subseteq I$ up to and including size c, there is some subset of $O'\subset O$ which has one-to-one correspondence with I' . Hence the capacity is c .

Q.E.D.

Example 2: The isolation vector of the concentrator of Figure 1(a) is (1,2,2).

$$H_1=h_1=1\leq 1 \quad \text{while} \quad H_2=h_1+h_2=1+2=3>2 .$$

Therefore the capacity of the concentrator is 2 .

By making use of Theorem 1 it is clearly straightforward to calculate the capacity of a concentrator when the isolation vector or the values of the isolation numbers $H_j, 1\leq j\leq |O|$, are known. But, in general, this information is not easy to obtain. However, for some special cases, such as binomial concentrators, we can give the following.

Lemma 1: The isolation vector of a $\binom{x}{y}$-concentrator is

$$\left(\binom{0}{y-1},\binom{1}{y-1},\ldots\binom{x-2}{y-1},\binom{x-1}{y-1}\right) .$$

Proof: An $\binom{x}{y}$-concentrator has $\binom{x}{y}$ input lines, while an $\binom{x-1}{y}$-concentrator (which is contained in an $\binom{x}{y}$-concentrator) has one less output line than the $\binom{x}{y}$-concentrator and only $\binom{x-1}{y}$ input lines. Therefore at most $\binom{x-1}{y}$ input lines are isolated by $x-1$ output lines. That is $H_{x-1}=\binom{x-1}{y}$. Now from the formula $\binom{a}{b}=\binom{a-1}{b}+\binom{a-1}{b-1}$, we have that

$$h_x=H_x-H_{x-1}=\binom{x}{y}-\binom{x-1}{y}=\binom{x-1}{y-1} .$$

By using this equation recursively the isolation vector is obtained in descending order.

Q.E.D.

With Lemma 1, we can prove the following.

Theorem 2 (Masson [2]): The capacity of $\binom{x}{y}$-concentrator is $y+2$.

Proof: Since in a concentrator, the number of inputs is always larger than the number of outputs, it follows that $x \geq y+2$.

From Lemma 1 $H_{y+1}=\binom{y+1}{y}=y+1$,

and $H_{y+2}=\binom{y+2}{y}=\frac{y+1}{y}(y+2)$,

Note that $H_{y+1}=y+1<y+2<\frac{y+1}{y}(y+2)=H_{y+2}$

Hence, the capacity is $y+2$.

Q.E.D.

Composite Concentrators

Given two one-stage sparse crosspoint networks, say, $A:(I_1,O_1,u_1)$ and $B:(I_2,O_2,u_2)$, we define the composite one-stage sparse crosspoint as $A\cdot B$: $(I_1 x I_2, O_1 x O_2, u_{12})$ or more simply (I_{12},O_{12},u_{12}) where I_{12} is the Cartesian product of I_1 and I_2 and O_{12} is the Cartesian product of O_1 and O_2 , and where u_{12}: $I_1 x I_2 \rightarrow O_1 x O_2$ such that

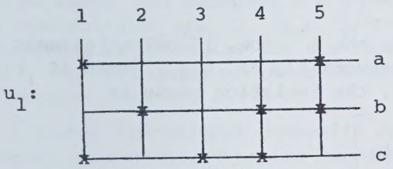
$$u_{12}(i_1 i_2)=\{u_1(i_1) x u_2(i_2)\}$$

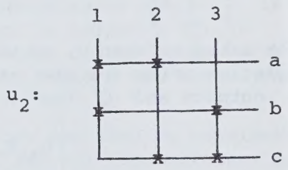
where $i_1 \varepsilon I_1$ and $i_2 \varepsilon I_2$ and $u_1(i_1) x u_2(i_2)$ is the Cartesian product of $u_1(i_1)$ and $u_2(i_2)$. The networks A and B will be referred to as the component networks of the composite network $A\cdot B$.

Example 3: Let A denote the network of Figure 1(a) and let B denote the network of Figure 1(c).

A: $(\{1,2,3,4,5\}, \{a,b,c\}, u_1)$

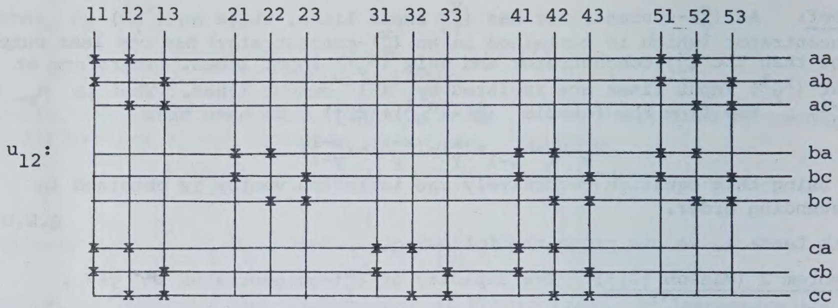
B: $(\{1,2,3\}, \{a,b,c\}, u_2)$





Then the composite network $A\cdot B$ has 15 inputs and 9 outputs and a composite relation u_{12}

$A\cdot B$: $(\{(1,1),(1,2),(1,3),(2,1),(2,2),(2,3),(3,1),(3,2),(3,3),(4,1),(4,2),(4,3),(5,1),(5,2),(5,3)\},\{(a,a),(a,b),(a,c),(b,a),(b,b),(b,c),(c,a)(c,b),(c,c)\},u_{12})$



Note that A•B is a composite concentrator. Hence if we knew the isolation vector of this composite concentrator, by Theorem 1 we could determine the capacity. However even though the isolation vectors of the components are known, the isolation vector of the components is not apparent. Hence, we will now define the concept of an isolation array with which to address this problem.

In general, suppose that two networks P and Q have following isolation vectors: $(p_1, p_2, \dots, p_n)$ and $(q_1, q_2, \dots, q_n)$. Then the isolation array for the composite network P•Q is an mxn array whose i^{th} row, j^{th} column element is equal to $p_i q_j$. In other words, it is the following vector product of isolation vectors:

$$\begin{bmatrix} p_1 \\ p_2 \\ \cdot \\ \cdot \\ \cdot \\ p_m \end{bmatrix} (q_1, q_2, \dots, q_n) = \begin{bmatrix} p_1q_1, & p_1q_2, & \dots., & p_1q_n \\ p_2q_1, & p_0q_2, & \dots., & p_2q_n \\ & --- & & \\ p_mq_1, & p_mq_2, & \dots., & p_mq_n \end{bmatrix}$$

Example 4: The isolation array of the composite network A•B of Example 3 is

$$\begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix} (0,1,2) = \begin{bmatrix} 0 & 1 & 2 \\ 0 & 2 & 4 \\ 0 & 2 & 4 \end{bmatrix} .$$

For later reference, we will denote the i^{th} row, j^{th} column element of the isolation array for the composite network P•Q as h_{ij} . Then if P has m outputs and Q has n outputs, the isolation array is

$$\begin{bmatrix} h_{11} h_{12} h_{13} \dots h_{1n} \\ h_{21} h_{22} h_{23} \dots h_{2n} \\ --- \\ h_{m1} h_{m2} h_{m3} \dots h_{mn} \end{bmatrix}$$

Capacity of Composite Concentrators

We will present in this section a theorem which is the basis of a capacity calculation technique for a class of composite concentrators. However, before doing this, we must first develop the concept of a smooth network.

We will define a network, say, P: (I,0,u) to be smooth if, given the network's isolation numbers $H_1, H_2, \ldots, H_{|0|}$, there exists a sequence of $|0|$ subsets of 0, denoted as $0_1, 0_2, \ldots, 0_{|0|}$, where

$$0_1 \subset 0_2 \subset 0_3 \subset \ldots \subset 0_{|0|}$$

and

$$|\sigma(0_j)| = H_j \; .$$

Example 5: The concentrator of Figure 1(a) (for which $H_1=1, H_2=3$, and $H_3=5$) is smooth as can be seen by examining $0_1=\{b\}$, $0_2=\{b,c\}$, and $0_3=\{a,b,c\}$ since $\sigma(\{b\})=\{2\}$, $\sigma(\{b,c\})=\{2,3,4\}$, $\sigma(\{a,b,c\}) = \{1,2,3,4,5\}$.

Finally, we will define an m(n)=partition of an integer c to be a partition of c into m integer parts, $r_1, r_2, \ldots, r_m$, such that $r_1+r_2+\ldots+r_m=c$, $0 \leq r_j \leq n$, and $r_i \geq r_j$ for $i<j$. We can now state the followong:

Theorem 3: Given the isolation array of the composite network P•Q where the isolation vector of P is $(p_1, \ldots, p_m)$ and the isolation vector of Q is $(q_1, \ldots, q_n)$ and at least one of these two component networks is smooth, then the capacity of P•Q is given by the minimum value of c which satisfies

$$\max_{\substack{\text{over all m(n)-partitions}\\ \text{of c}}} \left(\sum_{j=1}^{r_1} h_{1_j} + \sum_{j=1}^{r_2} h_{2_j} + \ldots + \sum_{j=1}^{r_m} h_{m_j} \right) > c \; .$$

The proof of Theorem 3 will be omitted in this version of this paper because of its length [3].

It should be noted that the complexity of calculating the capacity of a P•Q composite concentrator by repeatedly using Theorem 3 for increasing values of c can be shown to be $O(N^{3/2})$ [5].

Example 6: For the composite concentrator A•B of Example 3, the 3(3)-partition which satisfies Theorem 3 is: 3,3,0 . Hence, for the composite concentrator A•B, we have that c=6 . It should also be noted that using Theorem 3 repeatedly by starting with c=1 and programming to c=6 results in a consideration of 15 partitions.

Capacity Observations

Using Theorem 3, a wide variety of interesting capacity observations can be made. For example, a P•Q composite concentrator where P is a $\binom{u}{v}$-concentrator and Q is a $\binom{x}{y}$-concentrator is called a $\binom{u}{v}\binom{x}{y}$-concentrator. Formulas for the capacity in terms of the parameters u,v,x, and y for all cases of interest are listed in Table 1 and shown graphically in Figures 2(a)-2(d).

Higher dimensional composite concentrators can also be considered. For example, we can determine the capacity of a $\binom{4}{2}\binom{4}{2}\binom{4}{2}$-concentrator, or more simply, a $\binom{4}{2}^3$-concentrator using a 3-dimensional isolation matrix. Similarly $\binom{4}{2}^n$-concentrators can be considered. Table 2 gives the capacity of $\binom{x}{2}^n$-concentrators for $1 \leq n \leq 8$, $x \geq 4$.

Finally, Figure 3 shows a 3 dimensional isolation array for $\binom{u}{2}\binom{w}{3}\binom{y}{4}$-concentrators. Some rather interesting capacity observations for this 3 dimensional composite concentrator are the following: for $u \geq 4$, $w=4$, and $y=5$, $c=79$; for $u \geq 2$, $w=5$, and $y=5$, $c=74$; for $u \geq 2$, $u \geq 6$, and $y=5$, $c=58$; and for $u \geq 2$, $w \geq 4$, and $y \geq 6$, $c=48$.

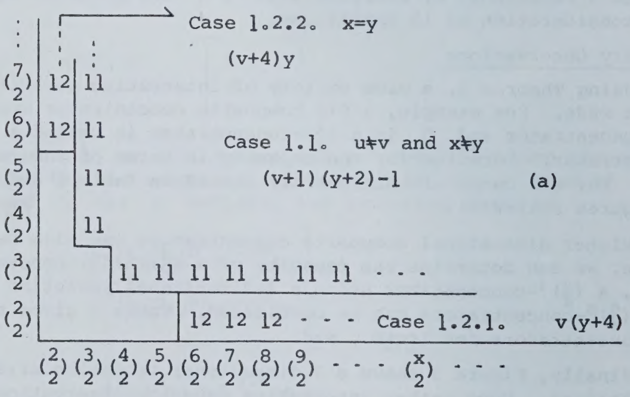
$\binom{u}{v}\binom{x}{y}$ - Concentrator $2\le v\le y$ and $\binom{u}{v}\binom{x}{y} > ux$

Case 1.	$y=2$		
Case 1.1.		$u\ne v$ and $x\ne y$	$c=(v+1)(y+2)-1$
Case 1.2.	$u=v$ or $x=y$		
Case 1.2.1.		$u=v$	$c=v(y+4)$
Case 1.2.2.		$x=y$	$c=(v+4)y$
Case 2.	$y\ne 2$		
Case 2.1.	$v=y$		
Case 2.1.1.		$u\le v+2$	$c=(v+1)(y+2)-1$
Case 2.1.2.		$u>v+2$	$c=v(y+3)$
Case 2.2.	$v\ne y$		
Case 2.2.1.		$x=y$	$c=uy$ where u is minimum and $\binom{u-1}{v-1}>vy$
Case 2.2.2.	$x=y+1$		
Case 2.2.2.1.		$2y>v(v+1)$	$c=(v+2)(y+1)$
Case 2.2.2.2.		$2y\le v(v+1)$	$c=(v+2)(y+1)-1$
Case 2.2.3.	$x=y+2$		
Case 2.2.3.1.		$2v>y$	$c=(v+1)(y+2)-1$
Case 2.2.3.2.		$2v\le y$	$c=v(y+2)$
Case 2.2.4.	$x\ge y+3$		
Case 2.2.4.1.		$2v>y$	$c=v(y+3)$
Case 2.2.4.2.		$2v\le y$	$c=v(y+2)$

Table 1

$\binom{u}{v}\binom{x}{y}$ - Concentrator $2\le v\le y$ and $\binom{u}{v}\binom{x}{y} > ux$

Case 1. $y=2$

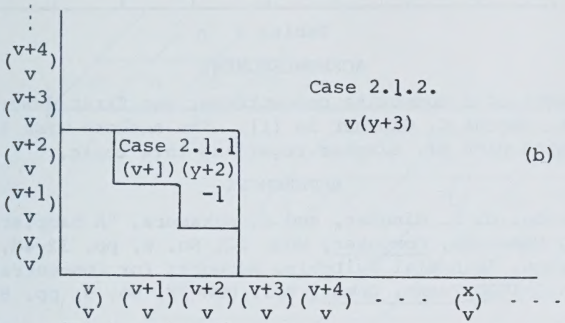


Case 1.2.2. x=y
(v+4)y
Case 1.1. u≠v and x≠y
(v+1)(y+2)-1
(a)
Case 1.2.1. v(y+4)
12
11
$\binom{7}{2}$
$\binom{6}{2}$
$\binom{5}{2}$
$\binom{4}{2}$
$\binom{3}{2}$
$\binom{2}{2}$
$\binom{2}{2}$ $\binom{3}{2}$ $\binom{4}{2}$ $\binom{5}{2}$ $\binom{6}{2}$ $\binom{7}{2}$ $\binom{8}{2}$ $\binom{9}{2}$ . . . $\binom{x}{2}$ . . .

Figure 2

Case 2. y≠2

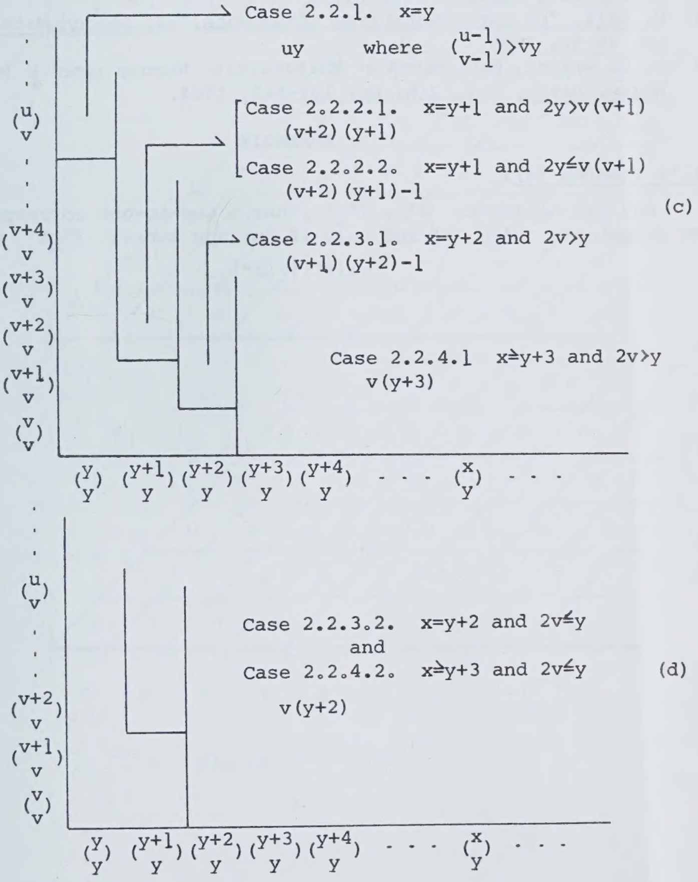
Case 2.1. v=y



$\binom{v+4}{v}$
$\binom{v+3}{v}$
$\binom{v+2}{v}$
$\binom{v+1}{v}$
$\binom{v}{v}$
Case 2.1.2.
v(y+3)
Case 2.1.1
(v+1)(y+2)
-1
(b)
$\binom{v}{v}$ $\binom{v+1}{v}$ $\binom{v+2}{v}$ $\binom{v+3}{v}$ $\binom{v+4}{v}$ . . . $\binom{x}{v}$ . . .

Case 2. y≠2 (continued)

Case 2.2. v=y



Case 2.2.1. x=y
uy where $\binom{u-1}{v-1}>vy$
$\binom{u}{v}$
Case 2.2.2.1. x=y+1 and 2y>v(v+1)
(v+2)(y+1)
Case 2.2.2.2. x=y+1 and 2y≤v(v+1)
(v+2)(y+1)-1
(c)
Case 2.2.3.1. x=y+2 and 2v>y
(v+1)(y+2)-1
Case 2.2.4.1 x≥y+3 and 2v>y
v(y+3)
$\binom{v+4}{v}$
$\binom{v+3}{v}$
$\binom{v+2}{v}$
$\binom{v+1}{v}$
$\binom{v}{v}$
$\binom{y}{y}$ $\binom{y+1}{y}$ $\binom{y+2}{y}$ $\binom{y+3}{y}$ $\binom{y+4}{y}$ . . . $\binom{x}{y}$ . . .
$\binom{u}{v}$
Case 2.2.3.2. x=y+2 and 2v≤y
and
Case 2.2.4.2. x≥y+3 and 2v≤y
v(y+2)
(d)
$\binom{v+2}{v}$
$\binom{v+1}{v}$
$\binom{v}{v}$
$\binom{y}{y}$ $\binom{y+1}{y}$ $\binom{y+2}{y}$ $\binom{y+3}{y}$ $\binom{y+4}{y}$ . . . $\binom{x}{y}$ . . .

Figure 2 (continued)

n	1	2	3	4	5	6	7	8
capacity	4	11	32	96	286	826	2588	7762

Table 2

ACKNOWLEDGMENT

The concept of a composite concentrator was first described by the authors and Mr. George C. Gingher in [1]. The authors wish to acknowledge useful discussion with Mr. Gingher regarding this topic.

REFERENCES

[1] G. M. Masson, G. C. Gingher, and S. Nakamura, "A Sampler of Circuit Switching Networks, *Computer*, Vol. 12, No. 6, pp. 32-48, June, 1979.

[2] G. M. Masson, "Binomial Switching Networks for Concentration and Distribution," *IEEE Trans. Comm.*, Vol. COM-25, No. 9, pp. 873-883, Sept., 1977.

[3] S. Nakamura and G. M. Masson, "Capacity Calculation for a Class of Composite Concentrators," Johns Hopkins University, Electrical Engineering Report 79-9, 1979.

[4] P. Hall, "On Representatives of Subsets," J. *London Math. Soc.*, Vol. 10, pp. 26-30, 1935.

[5] E. M. Wright, Partitions of Multipartite Number into k Parts, *J. Reine Angew. Math.*, Vol. 216, pp. 101-112, 1964.

APPENDIX

Hall's Theorem [4]:

Any set of inputs I', $I' \subseteq I$, has a one-to-one corresponsence with some output set $O' \subseteq O$ if and only if for any subset $I'' \subseteq I'$ we have that

$$|u(I'')| \geq |I''| \ .$$

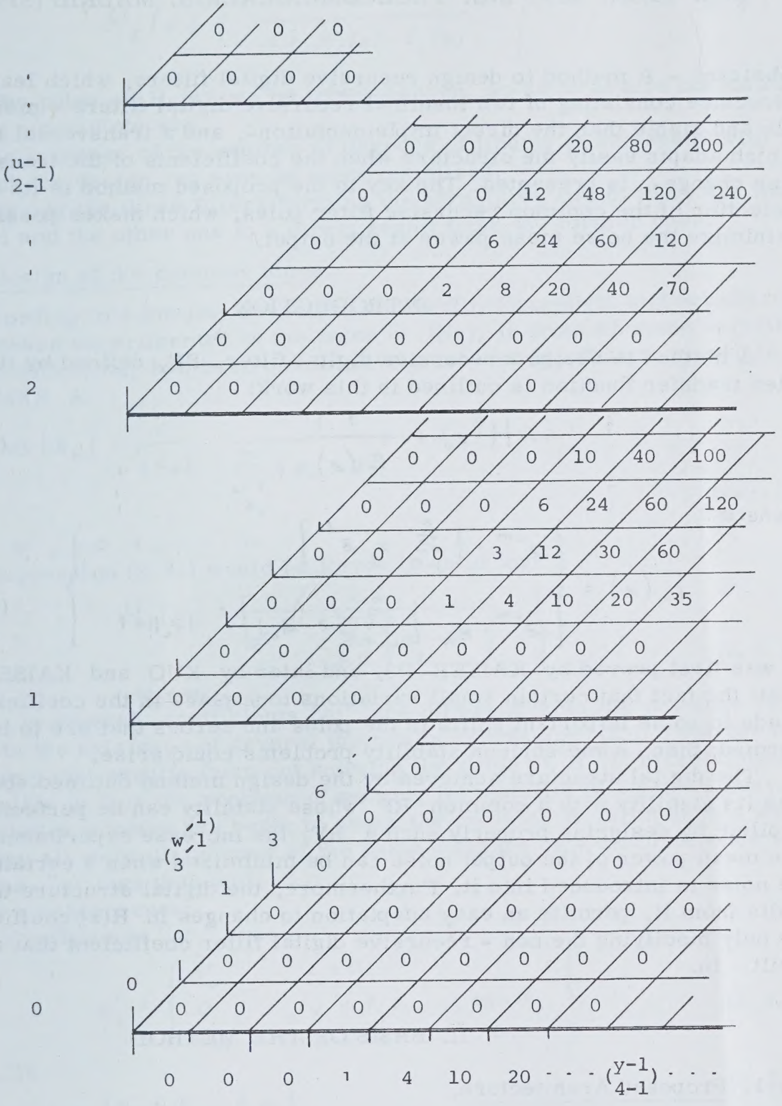

0 0 0
0 0 0 0
0 0 0 20 80 200
(u-1 2-1)
0 0 0 12 48 120 240
0 0 0 6 24 60 120
0 0 0 2 8 20 40 70
0 0 0 0 0 0 0 0
2
0 0 0 0 0 0 0 0
0 0 0 10 40 100
0 0 0 6 24 60 120
0 0 0 3 12 30 60
0 0 0 1 4 10 20 35
0 0 0 0 0 0 0 0
1
0 0 0 0 0 0 0 0
0 0 0 0 0 0
6
0 0 0 0 0 0 0
(w-1 3-1)
3
0 0 0 0 0 0 0
1
0 0 0 0 0 0 0 0
0
0 0 0 0 0 0 0 0
0
0
0 0 0 0 0 0 0 0
0 0 0 1 4 10 20 · · · (y-1 4-1) · · ·

Figure 3

A METHOD TO DESIGN RECURSIVE DIGITAL FILTERS

JESUS RIVERO LAGUNA
ESC. TECN. SUP. ING. TELECOMUNICACION. MADRID (SPAIN)

Abstract.- A method to design recursive digital filters, which leads to a structure consisting of two identical recursive digital filters - more simple and stable than the direct implementation-, and a transversal filter - which adapts easily the structure when the coefficients of the transfer function change-, is presented. The key to the proposed method is its original selection of the common recursive filter poles, which makes possible to minimize the noise mean power at the output.

I. INTRODUCTION

A method to design a recursive digital filter, RF, defined by its complex transfer function is outlined in this work:

$$H(z) = \frac{1}{E(z)} \tag{1.1.}$$

where :

$$E(z) = \left\{ \begin{array}{l} z^{-n} \cdot \left[\sum_{j=0}^{n} e_j z^j \right] \\ z^{-n} \cdot e_n \cdot \left[\prod_{i=1}^{n} (z - z_i) \right] , \quad |z_i| < 1 \end{array} \right\} \tag{1.2.}$$

It was first proved by KAISER (1), and later by KUO and KAISER (2), that: the fact that certain small variations took place in the coefficients, leads to some important shifts in the poles and zeroes that are to be performed; then, some serious stability problems could arise.

The digital structure achieved by the design method outlined above, assures its stability with a common RF whose stability can be perfectly controlled; by designing properly such a RF, the increase experimented by the mean power of the output noise can be minimized when a certain aditive noise is introduced into it. Furthermore, the digital structure that results from it, permits an easy adaptation to changes in H(z) coefficients, by only modifying the non - recursive digital filter coefficient that are built - in.

II. BASIS OF THE METHOD

II. 1. Proposed Architecture.

This method is based on the breakdown of E(z), following a completely general law:

$$E(z) \doteq E_1(z) + E_2(z) \tag{2.1.}$$

in which it is only required for all the zeroes of $E_1(z)$ to be inner to the unit circle of the complex plane ($|z|=1$). The structure that results from it, is shown in figure 1, with:

$$\left.\begin{aligned} H_1(z) &= \frac{1}{E_1(z)} \\ H_2(z) &= \frac{1}{1 + H_1(z)\cdot E_2(z)} \end{aligned}\right\} \qquad (2.2.)$$

where:

a) The poles of $H_1(z)$ can be freely chosen, so as to assure its stability in a complete way.

b) An increase of the number of degrees of liberty is achieved, with which the design can even be more optimized.

c) Among the three filters forming the structure, two of them are identical and the other one is a direct result from the former ones.

II. 2. Design of the common block.

According to a similar reasoning adopted by Oppenhein and Schaffer(3), the increase experimented in the poles of H(z), in front of some variations of the coefficients, has been quantitatively calculated. This is proved in APPENDIX A:

$$\left.\begin{aligned} \Delta |z_k| &= \frac{1}{|e_n|}\quad \frac{1}{\prod\limits_{\substack{i=1\\ i\neq j}}^{n} |z_k - z_j|}\cdot |z_k|^j \cdot |\Delta e_j| \\ \forall\, j &= 0, 1, \ldots, n \end{aligned}\right\} \qquad (2.3.)$$

Even expression (2. 3.) would be formally valid for:

$$\left.\begin{aligned} H_1(z) &= \frac{1}{z^{-m}\cdot \sum\limits_{j=0}^{m}\left[e'_j \cdot z^j\right]} \\ m &\leq n \end{aligned}\right\} \qquad (2.4.)$$

where, in general, coefficients "e'_i" are different from the previous "e_i".

Both the existence of several factors in expression (2. 3.) and the arbitrariety that results from not knowing the poles of $H_1(z)$ make almost impossible to face the problem of assuring its stability in a general way. The solution adopted has been: to chose the poles of $H_1(z)$ in such a way that they all were distributed at the same distance on a circunference with a radius "p" less than unit.

As a consequence:

a) Coefficients "e'_i" are:

$$e'_i = \left\{\begin{array}{ll} e'_0 & ,\ i = 0 \\ 0 & ,\ i = 1, 2, \ldots, m-1 \\ e'_m & ,\ i = m \end{array}\right\} \qquad (2.5.)$$

with

$$|e'_0| < |e'_m| \qquad (2.6.)$$

b) The increase that can be experimented by the module of the poles of $H_1(z)$ is:

$$\Delta P_{j=0,m} = \frac{1}{|e'_m|}\cdot \frac{1}{\prod\limits_{i=2}^{m} |z_i - z_1|}\cdot p^j\ |\Delta e'_j| \qquad (2.7.)$$

(pole z_1 has been considered as a reference of the set, because:

$$\prod_{\substack{i=1 \\ i \neq k}}^{m} |z_i - z_k| \quad = \text{constant}$$

independently from the k that is considered)

Furthermore, as it is proved in APPENDIX B, in this case:

$$\prod_{i=2}^{m} |z_i - z_1| = m \cdot P^{m-1} \tag{2.8}$$

We have as a result:

$$\Delta P_{j=0,m} = \frac{1}{|e'_m|} \cdot \frac{1}{m} \cdot P^{j-m+1} \cdot |\Delta e'_j| \tag{2.9}$$

II. 3. Common block design optimization.

It is deduced from expression (2.9) that:

a) It is a matter of interest for "e'_0" not to be changeable, because the contribution with a null j is very superior to the one that results when its value is m.

b) It is better for "e'_m" to be much higher than unit and, of course, it must verify (2.6).

c) It must be chosen the maximum possible value for m without exceeding the value of n.

According to these conclussions, design optimization of $H_1(z)$ requires for its transfer function to be:

$$H_1(z) = \frac{h_1 z^n}{z^n - h_2} \tag{2.10}$$

With:

$$\left.\begin{aligned} h_1 &= \frac{1}{e'_n} \\ h_2 &= -\frac{e'_0}{e'_n} \end{aligned}\right\} \tag{2.11}$$

The respective structure is shown in figure 2.

It has been proved through some studies made about h_1 and h_2 that:

a) h_2 will always be positive (in order to place z_1 in the positive real axis), although h_1 could be negative or positive, or what is the same:

$$\text{sgn}\left[e'_0\right] \neq \text{sgn}\left[e'_n\right] \tag{2.12}$$

b) h_1 would rather be much less than unit in its absolute value, although in general it could be higher; by the contrary, h_2 must always be less than unit.

c) Stability of $H_1(z)$ requires that:

$$\sqrt[n]{h_2} < \frac{n}{n + |h_1| \cdot |\Delta e'_n|} \tag{2.13}$$

as it can be proved from (2.9) (thinking about the physic sense of h_2, we can realize that the condition is not very hard).

d) If we take (2.14) the implementation of $H_1(z)$ turns to be slighty simplified:

$$/e'_0/ = 1 \tag{2.14}$$

III. PRACTIC ASPECTS

III. 1. Behaviour of the common block in front of an input noise.

Design of $H_1(z)$ has been optimized, on a basis of its behaviour in front of stationary and aditive input noise in a wide sense, with a mean power σ_i^2 and with a autocorrelation function:

$$R_n(t)\Big]_{t>T} = 0 \tag{3.1}$$

Then, the mean power rising from $H_1(z)$ would be:

$$\sigma_o^2 = \sigma_i^2 \sum_{r=0}^{\infty} h^2(rT) \tag{3.2}$$

Where $\{h(rT)\}$ are the samples of the filter impulsive response taken each T seconds:

$$\{h(rT)\} = \left\{ \begin{array}{ll} -\dfrac{h_2^{i+1}}{e'_0} & , \text{ for } \ r = n\cdot i \ (i = 0, 1, 2, \ldots) \\ 0 & , \text{ for the other ones} \end{array} \right\} \tag{3.3}$$

as it can be proved (4).

When substituting the previous $h(rT)$ values in expression (3.2)(as the progression tendency is assured by being $h_2 < 1$), it turns to be:

$$\frac{\sigma_o^2}{\sigma_i^2} = \left(\frac{1}{e'_0}\right)^2 \cdot \frac{h_2^2}{1 - h_2^2} \tag{3.4}$$

or even by (2.11):

$$\frac{\sigma_o^2}{\sigma_i^2} = \frac{1}{(e'_n)^2 - (e'_0)^2} \tag{3.5}$$

Even if we decide to simplify the implementation by making $/e'_0/$ equal to unit:

$$\frac{\sigma_o^2}{\sigma_i^2} = \frac{1}{(e'_n)^2 - 1} \tag{3.6}$$

Now then, it is inferred from the expression (3.6) that:

a) We are not concerned in the margin of values

$$1 < / e'_n / < \sqrt{2} \tag{3.7}$$

because the mean power of the input noise is notably increased.

b) It would rather be:

$$/ e'_n / > \sqrt{2} \tag{3.8}$$

For instance, next values could be chosen:

$$\left.\begin{array}{l} e'_0 = -1 \\ e'_n = 10 \end{array}\right\} \tag{3.9}$$

and this would permit to do:

$$h_1 = h_2 = 0'1 \tag{3.10}$$

and we would have as a consequence :

$$\left.\begin{aligned} P &= 10^{-n} << 1 \\ \Delta P &= 0 \\ \sigma_0^2 &= 10^{-2} \cdot \sigma_i^2 \end{aligned}\right\} \tag{3.11}$$

II. 2. Structure Adaptability.

According to the breakdown law, and taking (2.4) and (3.9) into account, it must be used a transversal filter as a non common block:

$$E_2(z) = z^{-n} \cdot \sum_{i=0}^{n} \left[e_i'' \, z^i \right] \tag{3.12}$$

with:

$$e_i'' = \left\{\begin{array}{ll} 1 + e_0 & , \quad i = 0 \\ e_n - 10 & , \quad i = n \\ e_i & , \quad \text{others} \end{array}\right\} \tag{3.13}$$

In general, this expression would be:

$$e_i'' = e_i - e_i' \tag{3.14}$$

Finally, the adaptation to changes in the value of coefficients defining H(z) involves the modification of the e_i'' corresponding to the $E_2(z)$ filter, while the whole structure and even $H_1(z)$ remain invariable.

IV. APLICATIONS

The previous method has been applied to digital equalizers (5). On the other hand, $H_1(z)$ filter can independently be used as a deemphasis digital filter in MF, being obvious both its correspondent preemphasis digital filter and the advantages concerned (6).

APPENDIX A

Increase calculation of the module of the poles.

According to (1.2):

$$H(z) = \left\{\begin{array}{l} f_1\,(e_0, e_1, \dots, e_n; z) \\ \\ f_2\,(e_n; z_1, z_2, \dots, z_n; z) \end{array}\right\} \tag{A.1}$$

where, obviously:

$$\left.\begin{array}{l} z_i = g_i\,(e_0, e_1, \dots, e_n) \\ \\ \forall_i = 1, 2, \dots, n \end{array}\right\} \tag{A.2}$$

Therefore:

$$\frac{\partial H(z)}{\partial e_j} = \sum_{r=1}^{n} \left[\frac{\partial H(z)}{\partial z_r} \cdot \frac{\partial z_r}{\partial e_j} \right] \tag{A.3}$$

$$\forall_j = 0,\ 1, 2, \ldots, n$$

By the individual calculation of the partial derivateds of H(z) and then replacing them in (A. 3), we can get:

$$\left.\begin{array}{l} -\dfrac{z^j}{e_n} = \displaystyle\sum_{r=1}^{n}\left[\prod_{\substack{i=1\\ i\neq r}}^{n}(z-z_i)\cdot\frac{\partial z_r}{\partial e_j}\right] \\ \forall_j = 0,\ 1, \ldots, n \end{array}\right\} \qquad (A.4)$$

Where, if we make it particular for a generic pole "z_k" (among all the possible "n") it turns to be:

$$\left.\begin{array}{l} \dfrac{\partial z_k}{\partial e_j} = -\dfrac{1}{e_n}\cdot\dfrac{z_k^j}{\displaystyle\prod_{\substack{i=1\\ i\neq j}}^{n}(z_k-z_i)} \\ \forall_j = 0, 1, \ldots, n \end{array}\right\} \qquad (A.5)$$

We can get the searched expression (2. 3) if last equation (A. 5) is staded in incremental terms.

APPENDIX B

Product calculation of the distances to reference pole.

We are concerned in calculating an expression equivalent to:

$$\prod_{i=2}^{m}|z_i - z_1| = \prod_{j=1}^{m-1} d_j \qquad (B.1)$$

where, as it can elementarily be probed:

$$d_j = |z_{j+1} - z_j| = 2P\cdot\left|\operatorname{sen}\left(\frac{\pi j}{m}\right)\right| \qquad (B.2)$$

Given the simetry of the poles set:

$$\prod_{j=1}^{m-1} d_j = \left\{\begin{array}{ll} d_1^2\cdot d_2^2\cdot \ldots \cdot d_{\frac{m-1}{2}}^2 , & \text{m odd} \\ d_1^2\cdot d_2^2\cdot \ldots \cdot d_{\frac{m-2}{2}}^2\cdot d_{\frac{m}{2}} , & \text{m even} \end{array}\right\} \qquad (B.3)$$

or by (B.2.):

$$\prod_{j=1}^{m-1} d_j = \left\{\begin{array}{ll} (2P)^{m-1}\cdot\displaystyle\prod_{j=1}^{\frac{m-1}{2}}\sin^2\left(\frac{\pi j}{m}\right) , & \text{m odd} \\ (2P)^{m-1}\cdot\displaystyle\prod_{j=1}^{\frac{m-2}{2}}\sin^2\left(\frac{\pi j}{m}\right) , & \text{m even} \end{array}\right\} \qquad (B.4)$$

Now then, it can be numerically proved, that when m is an odd number:

$$2^{m-1}\cdot\prod_{j=1}^{\frac{m-1}{2}}\sin^2\left(\frac{\pi j}{m}\right) = m \qquad (B.5)$$

and when m is an even number:

$$2^{m-1}\cdot\prod_{j=1}^{\frac{m-2}{2}}\sin^2\left(\frac{\pi j}{m}\right) = m \qquad (B.6)$$

therefore:

$$\prod_{j=1}^{m-1} d_j = m \cdot P^{m-1} \qquad \text{(B.7)}$$

REFERENCES

(1) - KAISER, J.F.
"Some practical considerations in the realization of linear digital filters"
PROC. 3rd. ALL. CONF. CIRC. SYST. TH:20-22, Oct. 1965;pg. 621-633.

(2) - KAISER, J.F.; KUO, FF.
"System analysis by digital computer"
JOHN-VILLEY; 1966.

(3) - OPPENHEIM, A.V.; SCHAFFER, R.W.
"Digital Signal Processing"
PRENTICE-HALL; 1975.

(4) - RIVERO LAGUNA, J.
"Incidencia del diseño y dimensionado de Filtros Transversales en la Igualación Digital de Canales Telefónicos".
Tes. Doct.: Univ. Pol. Madrid; 1977.

(5) - RIVERO LAGUNA, J.
"Una nueva filosofía de Igualación Digital Adaptativa, realizable con un miniordenador"
3rd WORLD TELECOMMUNICATION FORUM: Sept. 1979; Geneva (Switzerland).

(6) - RIVERO LAGUNA, J.
"Un nuevo método de síntesis de filtros para procesamiento digital de señales. Aplicación al diseño de filtros de preénfasis y deénfasis digitales".
SATELEC: 1977; Madrid (Spain).

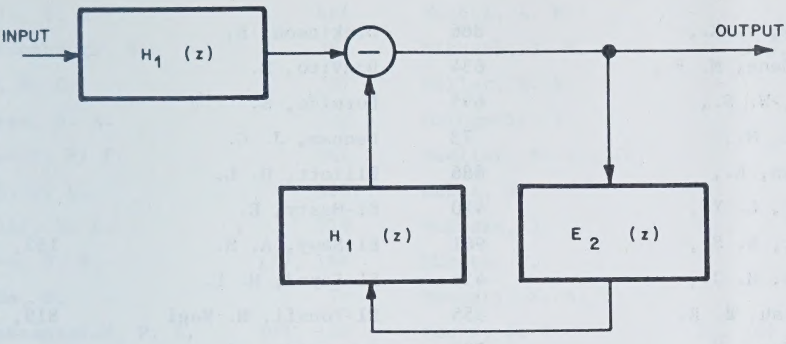

INPUT
H_1 (z)
—
OUTPUT
H_1 (z)
E_2 (z)

FIGURE 1. GENERAL STRUCTURE

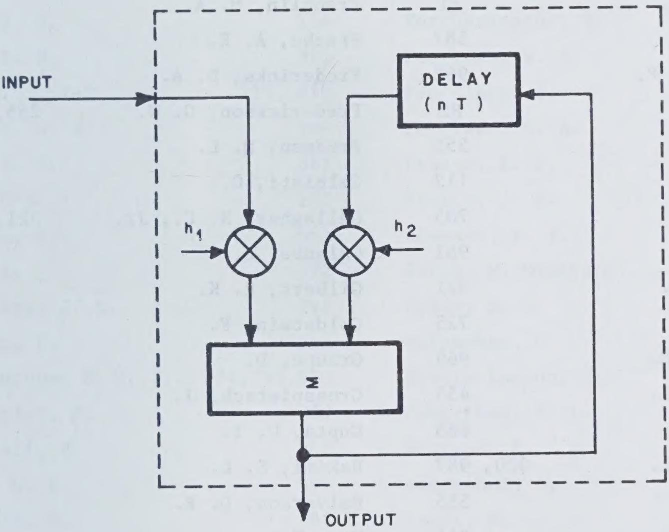

INPUT
DELAY
(nT)
h_1
h_2
Σ
OUTPUT

FIGURE 2. STRUCTURE OF $H_1(z)$

INDEX OF AUTHORS

Aatre, V. K., 886
Aburdene, M. F., 634
Agee, W. S., 695
Araki, M., 73
Ataman, E., 886
Bahar, L. Y., 410
Baker, B. S., 961
Balas, M. J., 434
Barmish, B. R. 555
Bar-Ness, Y. 787
Basar, T. 771
Bashir, A. 650
Bedrosian, S. D. 164
Belzer, M. R. 904
Benson, D. O., Jr. 606
Bentley, J. L. 11
Berman, N. 587
Bertsekas, D. P. 962
Bistritz, Y. 82
Blume, L. E. 555
Borth, D. E. 112
Bowles, W. M. 705
Brown, D. J. 961
Bucklew, J. A. 321
Calise, A. 725
Camerini, P. M. 969
Campbell, S. L. 455
Chang, C. B. 685
Chang, S. S. L. 920, 987
Chikte, S. D. 555
Chung, K. M. 940
Chyung, D. H. 798
Cohen, P. 896
Cooper, G. R. 420
Cornett, D. H. 624
David, A. J. 937
Dekel, E. 27
Dickinson, B. 596
DonVito, M. 133
Dormido, S. 206
Dunham, J. G. 802
Elliott, D. L. 540
El-Masry, E. 162
El-Sawy, A. H. 152, 866
El-Sayed, M. L. 102
El-Yousfi, M. Nagi 819, 830
Ephremides, A. 651
Evans, B. S. 110
Eyman, E. D. 840
Feinstein, J. 787
Fiduccia, C. M. 978
Fogel, E. 198
Franklin, M. A. 624
Frazho, A. E. 762
Fredericks, D. A. 735
Frederickson, G. N. 255, 470
Fredman, M. L. 231
Galbiati, G. 969
Gallagher, N. C., Jr. 321, 616
Gelenbe, E. 651
Gilbert, R. K. 284
Goldstein, F. 725
Graupe, D. 914
Grosspietsch, J. 914
Gupta, U. I. 225
Hakimi, S. L. 652
Halverson, D. R. 143
Han, Y. 122
Hart, J. M. 490
Helman, P. 509
Holt, C. S. 354
Hostetter, G. H. 215
Ikeda, M. 64
Inn, B. 162

Ja'Ja', J. 470, 677
Javid, S. H. 464
Johnsonbaugh, R. 387
Kao, M. C. 387
Kassam, S. A. 133
Katseff, H. P. 961
Keel, J. L. 996
Khalil, H. K. 779
Koivo, H. N. 179, 188
Kondo, B. 73
Krishnaprasad, P. S. 102, 110
Kwatny, H. G. 410
Kwong, R. H. S. 562
Landweber, L. H. 661
Lazar, A. A. 858
Lee, C. C. 848
Lee, D. T. 225
Lee, J. H. 164
Lee, T. N. 381
Leung, J. Y-T. 225, 659
Levine, W. S. 904
Lim, T. S. 381
Lin, C-S. 172
Lipton, R. 661
Liu, R. 172
LoCicero, J. L. 294
Luccio, F. 940
Mackenthun, K. M., Jr. 274, 293, 820
Maestrini, P. 329
Maffioli, F. 969
Maki, G. K. 996
Mallela, S. 364
Manacher, G. K. 949
Mark, J. W. 303
Massimo, F. 410
Masson, G. M. 364, 1016
McGillem, C. D. 579
Mehra, R. K. 712
Merrill, M. L. 659
Meyer, G. G. L. 350, 937
Michel, A. N. 546
Mikhail, S. H. 525, 533
Miller, R. K. 546
Mizoguchi, T. 265
Mueller, P. T., Jr. 1006
Mukai, H. 37
Mulawka, J. 374
Murata, T. 387
Muscati, Z. A. 810
Munro, J. I. 21, 241, 515
Nakajima, K. 652
Nakamura, S. 1016
Narasimhamurthi, N. 397
Nassimi, D. 1, 27
Neuhoff, D. L. 284
Olsder, G. J. 771
Parthasarathy, R. 339
Pearlman, W. A. 313
Penttinen, J. 179, 188
Pomalaza, C. A. 579
Prezas, P. P. 294
Pruitt, J. W. 225
Ramnath, R. V. 705
Rateb, M. Moustafa 819, 830
Reddy, S. M. 339, 349
Retnadhas, C. 640
Rivero Laguna, J. 1026
Robertson, E. L. 21, 515, 661
Root, W. L. 587
Rosenthal, A. 509
Saeki, M. 73
Sahni, S. 1, 27
Sain, M. K. 753
Sakla, A. A. 162
Salahi, J. 914
Savage, J. E. 670
Saxe, J. B. 480, 499
Scherzinger, B. M. 562

Schiavoni, N. 92
Schwartz, S. C. 858
de la Sen, M. 206
Seshadri, V. 753
Sezer, M. E. 54
Shaked, U. 82
Shapiro, E. Y. 735
Shiva, S. G. S. 810
Siegel, H. J. 1006
Siegel, L. J. 1006
Siljak, D. D. 54, 64
Simon, J. 677
Sira-Ramirez, H. 571
Skar, S. J. 403
Skelton, R. E. 421
Smith, J. E. 354
Sonnenschein, A. 596
Su, R. J. 540
Susarala, V. 650
Tabaczynski, J. A. 685
Tanttu, J. T. 179
Tarn, T. J. 540
Thomas, J. B. 848
Thowsen, A. 930
Todd, T. D. 303
Trick, T. N. 162
Tugnait, J. K. 876
Turner, R. H. 695
Tzou, K-H. 802
Vairavan, K. 650
Valette, A. 896
Van Scoy, F. L. 47
Varshney, P. K. 348
Vikmanis, M. 919
Wang, K. G. 920
Washburn, R. S., Jr. 712
Wei, K-C. 919
Weide, B. W. 245
Wise, G. L. 143, 616
Wong, C. K. 225, 940
Wong, K. M. 886
Young, K-K. D. 449
Zajac, F. E. 904
Zames, G. 744
Ziemer, R. E. 122
Zomlefer, M. R. 904

BOUND BY
HNM

UNIVERSITY OF ILLINOIS-URBANA
621.3811AL5 C001
PROCEEDINGS / ALLERTON CONFERENCE ON COM
1979
3 0112 008149145